



**PROYEK AKHIR - VE 180626**

**PENGENDALIAN NUTRISI DAN SISTEM INFORMASI KEBUN  
HIDROPONIK *NUTRIENT FILM TECHNIQUE* (NFT)  
MENGUNAKAN WEBSITE**

Gilang Sasmito  
NRP 10311600000042

Dosen Pembimbing  
Ir. Josaphat Pramudijanto, M.Eng.  
Fauzi Imaduddin Adhim, S.ST., M.T.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

-----Halaman ini sengaja dikosongkan-----



**FINAL PROJECT - VE 180626**

**NUTRIENT CONTROL AND SYSTEM INFORMATION  
NUTRIENT FILM TECHNIQUE (NFT) IN HYDROPONIC  
GARDEN WEBSITE BASED**

Gilang Sasmito  
NOR 10311600000042

*Supervisor*  
Ir. Josaphat Pramudijanto, M.Eng.  
Fauzi Imaduddin Adhim, S.ST., M.T.

*ELECTRICAL AND AUTOMATION ENGINEERING DEPARTMENT  
Vocational Faculty  
Sepuluh Nopember Institute of Technology  
Surabaya 2019*

-----Halaman ini sengaja dikosongkan-----

## PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul **“Pengendalian Nutrisi Sistem Informasi Kebun Hidropnik *Nutrient Film Technique* (NFT) Menggunakan Website”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2019



Gilang Sasmito  
NRP 10311500010042

-----Halaman ini sengaja dikosongkan-----

**PENGENDALIAN NUTRISI SISTEM INFORMASI KEBUN  
HIDROPNIK *NUTRIENT FILM TECHNIQUE* (NFT)  
MENGUNAKAN *WEBSITE***

**PROYEK AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Ahli Madya Teknik  
Pada  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui:



Ir. Josaphat Pramudijanto, M.Eng  
NIP. 19621005 199003 1 003

Pembimbing 2

A handwritten signature in black ink, belonging to Fauzi Imaduddin Adhim.

Fauzi Imaduddin Adhim, S.ST., M.T  
NPP. 1991201711057

**SURABAYA  
JULI, 2019**

-----Halaman ini sengaja dikosongkan-----



**PENGENDALIAN NUTRISI DAN SISTEM INFORMASI  
KEBUN HIDROPONIK *NUTRIENT FILM TECHNIQUE* (NFT)  
MENGUNAKAN *WEBSITE***

**Nama Mahasiswa** : Gilang Sasmito  
**NRP** : 10311600000042  
**Dosen Pembimbing 1** : Ir. Josaphat Pramudijanto, M.Eng  
**NIP** : 19621005 199003 1 003  
**Dosen Pembimbing 1** : Fauzi Imaduddin Adhim, S.ST., M.T  
**NPP** : 1991201711057

**ABSTRAK**

Hidroponik *Nutrient Film Technique* (NFT) merupakan sebuah sistem hidroponik yang banyak dikembangkan. Hidroponik NFT dikategorikan sistem hidroponik tertutup. Pada sistem hidroponik tertutup, air bersirkulasi selama 24 jam terus menerus. Namun untuk saat kadar nutrisi pada air sebagai media utama tanaman belum dapat dipantau dan dikendalikan secara otomatis.

Pada Proyek Akhir ini dibuatlah sebuah alat yang mampu mengatur kadar nutrisi sesuai setpoint yang telah ditentukan. Pengendalian kadar nutrisi dilakukan oleh STM32. Informasi hasil pengukuran sensor akan dikirimkan ke NodeMCU. Kemudian NodeMCU bertugas mengunggah data pengukuran sensor pada *database*. Sehingga kondisi pada air dapat ditampilkan pada *website*.

Berdasarkan hasil pengujian keseluruhan sistem didapatkan bahwa pengendalian nutrisi pada air kebun hidroponik dapat dilakukan dengan tepat dengan persentase kesalahan terbesar hanya 2,3%. Dan *website* sebagai sistem informasi mampu menampilkan data hasil pengukuran sensor, serta mampu melakukan pengendalian dengan memberikan nilai *setpoint*.

Kata kunci : Hidroponik, Pengendalian nutrisi hidroponik, sistem informasi hidroponik, STM32, NodeMCU.

-----Halaman ini sengaja dikosongkan-----

***NUTRIENT CONTROL SYSTEM INFORMATION NUTRIENT  
FILM TECHNIQUE (NFT) IN HYDROPONIC GARDEN WEBSITE  
BASED***

***Student Name*** : Gilang Sasmito  
***Registration Number*** : 10311600000042  
***Supervisor 1*** : Ir. Josaphat Pramudijanto, M.Eng  
***ID*** : 19621005 199003 1 003  
***Supervisor 2*** : Fauzi Imaduddin Adhim, S.ST., M.T  
***ID*** : 1991201711057

***ABSTRACT***

*Hydroponic Nutrient Film Technique (NFT) is a developed hydroponic system. NFT hydroponics is a closed hydroponic system. In closed hydroponic systems, water circulates for 24 hours continuously. But for when the level of nutrients in water as the main medium of plants can not be monitored and controlled automatically.*

*In this Final Project a tool was made that was able to regulate nutrient levels according to a predetermined setpoint. Nutrition control is carried out by STM32. Information on measuring the sensor will be sent to NodeMCU. Then NodeMCU is tasked with uploading sensor measurement data to the database. So that conditions on water can be displayed on the website.*

*Based on the results of testing the entire system, it was found that the control of nutrients in hydroponic garden water can be carried out correctly with the largest percentage of errors only 2.3%. And the website as an information system is able to display sensor measurement data, and be able to control by giving a setpoint value.*

***Keywords:*** *Hydroponics, hydroponic nutrition control, hydroponic information system, STM32, NodeMCU.*

-----Halaman ini sengaja dikosongkan-----

## KATA PENGANTAR

Alhamdulillah kami panjatkan kepada Allah Subhanahu Wa Ta'ala, atas limpahan rahmat dan kemudahan dariNya, hingga kami dapat menyelesaikan Proyek Akhir ini dengan baik, begitu pula dengan pembuatan buku Proyek Akhir ini.

Proyek Akhir ini dilakukan untuk memenuhi beban satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Departemen Teknik Elektro Otomasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya untuk menyelesaikan program pendidikan Diploma di Teknik Elektro dengan judul :

### **“Sistem Informasi Kebun Hidroponik *Nutrient Film Technique* (NFT) Menggunakan *Website*”**

Dengan terselesainya Proyek Akhir ini Penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Kedua orang tua yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Ir. Joko Susila, M.T. selaku Kepala Departemen Teknik Elektro Otomasi
3. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku dosen pembimbing pertama dengan segala bimbingan ilmu, moral, dan spiritual.
4. Bapak Fauzi Imaduddin Adhim, S.ST., M.T. selaku dosen pembimbing kedua dengan segala bimbingan ilmu, moral, dan spiritual.
5. Teman - teman D3 Teknik Elektro yang selalu memberikan doa, bantuan, semangat, dan dukungannya.
6. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhir kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juni 2019

Gilang Sasmito

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

## HALAMAN

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN PROYEK AKHIR ..... <b>Error! Bookmark not defined.</b>	
LEMBAR PENGESAHAN ..... <b>Error! Bookmark not defined.</b>	
ABSTRAK.....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xix
DAFTAR TABEL.....	xxiii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	2
1.3 Tujuan .....	2
1.4 Sistematika Laporan.....	2
1.5 Relevansi.....	3
<b>BAB II LANDASAN TEORI .....</b>	<b>5</b>
2.1 Hidroponik .....	5
2.2 <i>Part Per Milion</i> (PPM) Pada Hidroponik.....	7
2.3 STM32 F446RE NUCLEO .....	8
2.4 Sensor TDS .....	8
2.5 Sensor Temperatur DS18B20 .....	9
2.6 Relay .....	10
2.7 Transistor .....	11
2.8 LCD 20 x 4.....	13
2.9 Modul <i>Stepdown</i> LM2596.....	14
2.10 NodeMCU ESP8266 .....	15
2.11 STM32 CubeMX.....	16
2.12 Atolic TrueSTUDIO.....	17
2.13 Arduino IDE.....	18
<b>BAB III PERANCANGAN PERANGKAT KERAS DAN</b>	
<b>PERANGKAT LUNAK .....</b>	<b>21</b>
3.1 Perancangan Sistem Keseluruhan .....	21

3.2	Perancangan Mekanik.....	22
3.2.1	Perancangan <i>Box Hardware</i> .....	22
3.2.2	Perancangan <i>Plant</i> .....	26
3.3	Perancangan Elektronik .....	28
3.3.1	Perancangan Subsistem Pengendali .....	29
3.3.2	Perancangan Subsistem Sensor.....	33
3.3.3	Perancangan Subsistem <i>Driver</i> .....	35
3.3.4	Perancangan Subsistem Aktuator .....	37
3.3.5	Perancangan Subsistem Catudaya.....	37
3.3.6	Perancangan Subsistem <i>Display</i> .....	38
3.3.7	Perancangan Subsistem <i>Internet of Thing (IoT)</i> .....	40
3.3.8	Perancangan <i>Board PCB</i> .....	42
3.4	Perancangan <i>Software</i> .....	43
3.4.1	Perancangan Program STM32F446RE.....	44
3.4.1.1	Pembacaan Sensor .....	46
3.4.1.2	Pengendalian Aktuator.....	48
3.4.1.3	Serial Komunikasi Dengan NodeMCU ....	50
3.4.2	Pemrograman NodeMCU .....	51
3.4.2.1.	Serial Komunikasi Dengan STM32 F446RE.....	51
3.4.2.2.	Pengiriman dan Penerimaan Data Pada <i>Database</i> .....	52
3.4.3	Perancangan <i>Website</i> Sebagai Sistem Informasi ....	53
3.4.3.1	Perancangan Tampilan <i>Website</i> .....	54
3.4.3.3	Menampilkan Data Dari <i>Database</i> Pada <i>Website</i> .....	57
BAB IV PENGUKURAN DAN ANALISA .....		61
4.1	Hasil Perancangan <i>Hardware</i> .....	61
4.1.1	Hasil Perancangan Mekanik .....	61
4.1.2	Hasil Perancangan Elektronik.....	62
4.2	Pengujian Subsistem .....	62
4.2.1	Pengujian <i>Setup</i> Subsistem Pengendali .....	62
4.2.2	Pengujian Subsistem Sensor .....	65
4.2.3	Pengujian Subsistem <i>Driver</i> .....	69
4.3	Pengambilan Data Pemberian Nutrisi .....	74
4.3.1	Pemberian Nutrisi Pada Air PDAM Surabaya .....	74
4.3.2	Pemberian Nutrisi Pada Air Destilasi .....	76
4.3.3	Pemberian Nutrisi Pada Air Pandaan.....	79



4.3.4	Pemberian Nutrisi Pada Air Pembuangan Air Coinditioner (AC) .....	81
4.3.5	Pemberian Nutrisi Pada Air Mineral .....	83
4.4	Pengujian Ketepatan Pengendalian Nutrisi .....	85
4.4.1	Pengujian Pengaturan Nutrisi pada Air Pandaan.....	85
4.4.2	Pengujian Pengaturan Nutrisi pada Air PDAM Surabaya.....	86
4.4.3	Pengujian Pengaturan Nutrisi pada Air Destilasi ....	87
4.4.4	Pengujian Pengaturan Nutrisi pada Air Mineral .....	88
4.5	Pengujian Website Sebagai Sistem Informasi .....	89
4.5.1	Pengujian Ketepatan Autentifikasi Pada Login Page .....	89
4.5.2	Menampilkan Data Sensor Pada Website .....	92
4.5.3	Penulisan Data Pada Database Melalui Website .....	93
4.6	Pengujian Keseluruhan Sistem.....	94
4.6.1	Pengujian Login User.....	95
4.6.2	Pengendalian Melalui Website. ....	95
4.6.3	Pengujian Sistem Pada Hidroponik.....	95
BAB V PENUTUP.....		98
5.1	Kesimpulan .....	99
5.2	Saran .....	99
DAFTAR PUSTAKA .....		101
LAMPIRAN A .....		A-1
LAMPIRAN B .....		B-1
LAMPIRAN C .....		C-1
DAFTAR RIWAYAT HIDUP		

-----Halaman ini sengaja dikosongkan-----

## DAFTAR GAMBAR

### HALAMAN

Gambar 2. 1 Hidroponik dengan Sistem NFT .....	6
Gambar 2. 2 Hidroponik dengan Sistem <i>Ebb and Flow</i> .....	6
Gambar 2. 3 Hidroponik dengan Sistem <i>Floating Raft</i> .....	7
Gambar 2. 4 Hidroponik dengan Sistem <i>Floating Raft</i> .....	7
Gambar 2. 5 <i>Board</i> STM32 F466RE NUCLEO .....	8
Gambar 2. 6 Probe dan Modul Sensor TDS.....	9
Gambar 2. 7 Bentuk Fisik Sensor DS18B20.....	10
Gambar 2. 8 Pin Konfigurasi Relay dan Kondisi NO dan NC Relay	11
Gambar 2. 9 Bentuk Fisik, Simbol dan Pin Kofigurasi Transistor ....	12
Gambar 2. 10 LCD 20x4.....	14
Gambar 2. 11Modul <i>Stepdown</i> LM2596.....	15
Gambar 2. 12 <i>Board</i> NodeMCU ESP8266 .....	16
Gambar 2. 13Tampilan STM32 CubeMX.....	17
Gambar 2. 14 Tampilan Muka Atolic TrueStudio .....	17
Gambar 2. 15 <i>Sketch</i> dari <i>Software</i> Arduino IDE .....	18
Gambar 2. 16 Tampilan Atom Editor.....	19
Gambar 3. 1 Diagram Fungsional Sistem .....	21
Gambar 3. 2 Desain Mekanik Sistem.....	22
Gambar 3. 3 Desain <i>Box Hardware</i> .....	23
Gambar 3. 4 Dimensi Sisi Depan <i>Box</i> .....	24
Gambar 3. 5 Dimensi Sisi Tengah <i>Box</i> .....	25
Gambar 3. 6 Dimensi Sisi Samping <i>Box</i> .....	25
Gambar 3. 7 Dimensi Sisi Bawah <i>Box</i> .....	26
Gambar 3. 8 Dimensi Sisi Bawah <i>Box</i> .....	26
Gambar 3. 9 Desain <i>Plant</i> Kebun Hidroponik .....	27
Gambar 3. 10 Dimensi <i>Plant</i> Media Tanaman.....	28
Gambar 3. 11 Dimensi Wadah Penampungan Air .....	28
Gambar 3. 12 Alokasi Pin-Pin pada pada STM32 F446RE NUCLEO .....	30
Gambar 3. 13 Memilih <i>Board</i> Pada STM32 CubeMX .....	31
Gambar 3. 14 Konfigurasi Pin STM32 pada Sistem .....	32
Gambar 3. 15 <i>Clock</i> Konfigurasi STM32 .....	32
Gambar 3. 16 <i>Wiring</i> Diagram DS18B20 dengan STM32 F446RE .	33

Gambar 3. 17 <i>Wiring Diagram</i> Sensor TDS dengan STM32 F446RE NUCLEO .....	33
Gambar 3. 18 <i>Wiring Diagram</i> DS18B20 dengan STM32 F446RE .....	34
Gambar 3. 19 Diagram Koneksi Sensor Temperatur DS18B20 dengan STM32 F446RE NUCLEO .....	35
Gambar 3. 20 Skematik <i>Driver</i> Aktuator Pompa Peristaltik.....	36
Gambar 3. 21 Koneksi Rangkaian <i>Driver</i> Dengan Pompa Peristaltik.....	37
Gambar 3. 22 Rangkaian Catu Daya dengan Subsistem lain .....	38
Gambar 3. 23 <i>Wiring</i> Skematik LCD20x4 dengan STM32 F446RE.....	39
Gambar 3. 24 Diagram Koneksi LCD20x4 Dengan STM32 .....	39
Gambar 3. 25 <i>Wiring</i> Skematik NodeMCU dengan STM32 F446RE.....	41
Gambar 3. 26 Diagram Koneksi STM32 dengan NodeMCU .....	41
Gambar 3. 27 Skematik PCB Pengendali Menggunakan <i>Software Eagle Pro</i> .....	42
Gambar 3. 28 Skematik PCB Sensor dan <i>Driver</i> menggunakan <i>software Eagle Pro</i> .....	43
Gambar 3. 29 Diagram Alir Sistem .....	45
Gambar 3. 30 <i>Flowchart</i> Pembacaan Sensor .....	46
Gambar 3. 31 Inisialisasi Sensor DS18B20 .....	47
Gambar 3. 32 <i>Master Write</i> “1” <i>Slot</i> .....	47
Gambar 3. 33 <i>Master Write</i> “0” <i>Slot</i> .....	48
Gambar 3. 34 <i>Flowchart</i> Pengendalian Aktuator .....	49
Gambar 3. 35 <i>Flowchart</i> Pengiriman Data STM32 F446RE ke NodeMCU .....	50
Gambar 3. 36 <i>Flowchart</i> Penerimaan Data STM32 F446RE Oleh NodeMCU .....	50
Gambar 3. 37 <i>Flowchart</i> Pengiriman Data Pada <i>Database</i> Melalui NodeMCU .....	52
Gambar 3. 38 <i>Flowchart</i> Pembacaan Data Pada <i>Database</i> Oleh NodeMCU .....	53
Gambar 3. 39 Desain Tampilan <i>Website</i> .....	54
Gambar 3. 40 Desain Tampilan <i>Login Page</i> .....	54
Gambar 3. 41 <i>Flowchart</i> Proses Pada <i>Login</i> .....	56
Gambar 3. 42 <i>Flowchart</i> Pembacaan Data Pada <i>Database</i> .....	57
Gambar 3. 43 <i>Flowchart</i> Pengiriman Data ke <i>Database</i> Melalui <i>Website</i> .....	58
Gambar 4. 1 Hasil Perancangan Mekanik.....	61

Gambar 4. 2 Desain <i>Layout</i> PCB .....	62
Gambar 4. 3 Program Pin <i>Set</i> Pada STM32 .....	63
Gambar 4. 4 Program Pin <i>Reset</i> Pada STM32 .....	63
Gambar 4. 5 Rangkaian Pengukuran Tegangan Pin STM32.....	63
Gambar 4. 6 Program Pembacaan ADC.....	64
Gambar 4. 7 Rangkaian Pengujian <i>Chanel</i> ADC pada STM32 .....	64
Gambar 4. 8 Program Pembacaan Sensor TDS.....	66
Gambar 4. 9 Pengukuran dengan Sensor TDS dan TDS Meter .....	66
Gambar 4. 10Program Pembacaan Sensor DS18B20 .....	68
Gambar 4. 11 Pengukuran Sensor DS18B20 dan Termometer .....	68
Gambar 4. 12 Pengujian Rangkaian <i>Driver</i> .....	70
Gambar 4. 13 Pengujian Debit Pompa Peristaltik.....	71
Gambar 4. 14 Program LCD Pengujian Bit .....	73
Gambar 4. 15 Tampilan LCD Pengujian Bit.....	73
Gambar 4. 16 Tampilan LCD Sesuai dengan Inisial Kondisi Awal Sensor .....	73
Gambar 4. 17 Tampilan LCD Sesuai dengan Hasil Pengukuran .....	74
Gambar 4. 18Memasukan Data Yang Salah Pada <i>Form Login</i> .....	90
Gambar 4. 19 Error Karena Data <i>Login</i> Salah .....	90
Gambar 4. 20 Memasukan Data Yang Salah Pada <i>Form Login</i> .....	91
Gambar 4. 21 Halaman Utama <i>Website</i> .....	91
Gambar 4. 22 Data Pada <i>Database</i> .....	92
Gambar 4. 23 Tampilan <i>Dashboard Website</i> .....	93
Gambar 4. 24 <i>Form Setup</i> pH .....	93
Gambar 4. 25 <i>Form Setup</i> TDS.....	94
Gambar 4. 26 Data Pada <i>Database</i> Setelah Dilakukan <i>Setup</i> TDS dan pH .....	94
Gambar 4. 27 Tampilan <i>Website</i> Sebelum Pengendalian.....	95

-----Halaman ini sengaja dikosongkan-----

## DAFTAR TABEL

	HALAMAN
Tabel 2. 1 Data Pin LCD 20x4 .....	13
Tabel 2. 2 Spesifikasi NodeMCU .....	15
Tabel 3. 1 Letak Komponen Pada <i>Box Hardware</i> .....	23
Tabel 3. 2 Pin Konfigurasi Sensor TDS dengan STM32 F446RE NUCLEO .....	34
Tabel 3. 3 Pin Konfigurasi Sensor Temperatur DS18B20 dengan STM32 F446RE NUCLEO .....	35
Tabel 3. 4 Komponen Pembuatan <i>Driver</i> .....	36
Tabel 3. 5 Pin Konfigurasi Sensor Temperatur DS18B20 dengan STM32 F446RE NUCLEO .....	40
Tabel 3. 6 Pin Konfigurasi Sensor TDS dengan STM32 F446RE NUCLEO .....	42
Tabel 3. 7 Tabel <i>Software</i> yang Digunakan .....	44
Tabel 4. 1 Data Pengukuran Tegangan Pin STM32 .....	63
Tabel 4. 2 Perbandingan Pembacaan ADC STM32 dengn Perhitungan .....	64
Tabel 4. 3 Perbandingan Pengukuran Sensor TDS dengan TDS Meter .....	66
Tabel 4. 4 Perbandingan Pengukuran Sensor Temperatur DS18B20 dengan Termometer .....	68
Tabel 4. 5 Perbandingan <i>VBB</i> Dengan Kondisi Relay.....	70
Tabel 4. 6 Data Pengukuran Volume Air dan Waktu.....	72
Tabel 4. 7 Data TDS Pemberiaan Nutrisi Pada Air PDAM Surabaya.....	74
Tabel 4. 8 Data TDS Pemberiaan Nutrisi Pada Air Destilasi .....	77
Tabel 4. 9 Data TDS Pemberiaan Nutrisi Pada Air Pandaan .....	79
Tabel 4. 10 Data TDS Pemberiaan Nutrisi Pada Air AC .....	81
Tabel 4. 11 Data TDS Pemberiaan Nutrisi Pada Air Mineral .....	83
Tabel 4. 12 Perbandingan <i>Setpoint</i> dengan TDS Akhir Pada Air Pandaan .....	85
Tabel 4. 13 Perbandingan <i>Setpoint</i> dengan TDS Akhir Pada Air .....	86
Tabel 4. 14 Perbandingan <i>Setpoint</i> dengan TDS Akhir Pada Air Destilasi	87

Tabel 4. 15 Perbandingan *Setpoint* dengan TDS Akhir Pada Air Mineral.....88



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dewasa ini kota – kota besar di Indonesia mulai kehilangan lahan sebagai media tanam. Hal ini disebabkan karena tingkat pembangunan Indonesia yang sangat tinggi dibuktikan dengan data *Global Competitiveness Index* pada tahun 2017 yang menunjukan bahwa indeks daya saing Indonesia pada 2017-2018 berada di urutan ke-52 dari posisi sebelumnya di periode 2015-2016 yang masih berada di posisi 62. Berkurangnya lahan pertanian juga terjadi salah satu kota terbesar di Indonesia yaitu Surabaya. Tercatat bahwa pada tahun 2015 tersedia 1.700 hektar (ha) area pertanian namun pada tahun 2016 tersisa 1.400 ha, pada kurun waktu satu tahun luas lahan pertanian menyusut 300 ha. Namun dengan seiringnya perkembangan teknologi, sektor pertanian juga ikut mengalami perkembangan.

Salah satu perkembangannya adalah pengembangan pola cocok tanam tanpa media tanah. Pola cocok tanam ini dikenal dengan nama Hidroponik. Hidroponik berasal dari kata *hydro* yang berarti air dan *ponos* yang berarti daya. Dengan demikian hidroponik dapat diartikan sebagai memberdayakan air.

Pola cocok tanam ini juga telah diterapkan oleh tempat magang saya. Pada tempat magang saya menggunakan metode untuk pola cocok tanam yang digunakan adalah sistem *Nutrient Film Technique* (NFT). Dimana NFT merupakan model budidaya dengan meletakkan akar tanaman pada lapisan air yang dangkal. Air tersebut tersirkulasi dan mengandung nutrisi sesuai kebutuhan tanaman. Perakaran bisa berkembang didalam larutan nutrisi. Karena disekeliling perakaran terdapat lapisan larutan nutrisi maka sistem ini dikenal dengan nama *nutrient film technique*. Selain nutrisi yang dibutuhkan adalah temperature air yang baik juga mempengaruhi tingkat pertumbuhan dan kesehatan tanaman. Sehingga harus dilakukan pengukuran dan pengaturan kandungan nutrisi air atau *Total Dissolved Solid* (TDS) dan temperatur. Dilain sisi pada tempat magang saya pengukuran dan pengaturan masih dilakukan secara manual. Sehingga dalam satu hari hanya dapat dilakukan pengukuran dan pengaturan dua kali yaitu saat pagi hari dan siang hari. Sehingga diperlukan suatu alat yang mampu mengendalikan nutrisi pada air kebun hidroponik secara otomatis agar nutrisi pada air dapat terjaga

dan selalu konstan. Dan dibutuhkan sebuah media yang mampu menampilkan kondisi air seperti kondisi pH, kadar nutrisi, temperatur serta *level* air yang ada pada bak penampungan air.

## **1.2 Permasalahan**

Perumusan masalah yang dibahas dalam pengerjaan Proyek Akhir ini adalah sebagai berikut:

1. Nutrisi pada air kebun hidroponik belum terkendali dengan baik.
2. Tidak mengetahuia pada kondisi apa air pada kebun hidroponik harus diganti.
3. Belum adanya media yang mampu memberikan informasi kondisi air pada kebun hidroponik.
4. Belum adanya peringatan untuk penggantian air kebun hidroponik karena buruknya kualitas air.

## **1.3 Tujuan**

Tujuan yang dicapai dalam penyelesaian Proyek Akhir ini adalah sebagai berikut:

1. Mampu mengendalikan kadar nutrisi pada air kebun hidroponik sesuai dengan *setpoint*.
2. Mengetahui waktu penggantian air kebun hidroponik berdasarkan kualitas air.
3. Menyajikan informasi mengenai kondisi air pada kebun hidroponik menggunakan *website*

## **1.4 Sistematika Laporan**

Pembahasan Poyek Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

### **Bab I Pendahuluan**

Bab ini meliputi latar belakang, permasalahan, batasan masalah, tujuan, sistematika laporan, dan relevansi.

### **Bab II Teori Dasar**

Bab ini membahas teori dasar, teori penunjang dari peralatan-peralatan yang digunakan dalam pembuatan alat.

### **Bab III Perancangan Sistem**

Bab ini membahas mengenai desain perancangan dan pembuatan perangkat keras (*hardware*) yang terdiri dari atas perancangan rangkaian elektronika, perancangan mekanik

serta perangkat lunak (*software*) yang terdiri atas program untuk mengelompokkan barang dan menghitung jumlah barang yang lewat pada konveyor.

#### **Bab IV Pengujian dan Analisa Sistem**

Bab ini membahas tentang pengukuran, pengujian alat dan analisa data yang didapat dalam pengujian alat.

#### **Bab V Penutup**

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

### **1.5 Relevansi**

Dengan dibuatnya sistem informasi ini diharapkan mampu membantu perusahaan tempat magang saya dalam memepertahankan dan memenuhi kebutuhan nutrisi pada tanaman hidroponik NFT serta mengembangkan teknologi pada bidang horikultura.

-----Halaman ini sengaja dikosongkan-----

## **BAB II**

### **LANDASAN TEORI**

Bab ini membahas mengenai teori penunjang dari peralatan yang digunakan dalam membuat sebuah pengendalian nutrisi dan sistem informasi pada kebun hidroponik berbasis *website*.

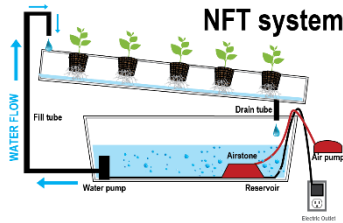
#### **2.1 Hidroponik [1]**

Hidroponik adalah suatu istilah yang digunakan untuk bercocok tanam tanpa menggunakan tanah sebagai media tumbuhnya. Tanaman dapat di tanam dalam pot atau wadah lainnya dengan menggunakan air dan atau bahan-bahan porous lainnya, seperti kerikil, pecahan genting, pasir, pecahan batu ambang, dan lain sebagainya sebagai media tanamnya. Bertanam secara hidroponik dapat berkembang secara cepat karena memiliki kelebihan. Kelebihan yang utama adalah keberhasilan tanaman untuk tumbuh dan berproduksi lebih terjamin. Kelebihan lainnya adalah perawatan lebih praktis, pemakaian pupuk lebih hemat, tanaman dapat tumbuh dengan pesat dan tidak kotor, hasil produksi lebih kontinu, serta beberapa jenis tanaman dapat dibudidayakan diluar musim.

Dalam melakukan budidaya tanaman hidroponik terdapat beberapa macam pola cocok tanam, diantaranya adalah:

##### **2.1.1 Hidroponik *Nutrient Film Technique* (NFT)**

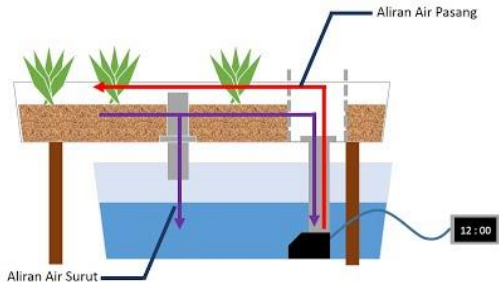
*Nutrient Film Technique* (NFT) merupakan model budi daya dengan meletakkan akar tanaman pada lapisan air yang dangkal. Air tersebut tersirkulasi dan mengandung nutrisi sesuai kebutuhan tanaman. Perakaran bisa berkembang di dalam larutan nutrisi. Karena di sekeliling perakaran terdapat selapis larutan nutrisi maka sistem ini dikenal dengan nama *nutrient film technique*. Pada hidroponik NFT, tanaman ditempatkan pada *stereofom* dengan akar menjuntai dibawahnya. *Stereofom* tersebut lalu ditempatkan pada sebuah talang yang dipasang dengan kemiringan 5% (turun 5 cm/m). Pada talang tersebut lalu dialirkan nutrisi setebal 3 - 4 mm secara terus - menerus (24 jam) ataupun berseling (dengan batas waktu maksimal tidak dialiri larutan selama 10 menit). Nutrisi ditempatkan dalam sebuah tandon (tempat penampungan). Nutrisi yang telah dialirkan ke dalam talang, dikembalikan lagi ke dalam tandon.



Gambar 2. 1 Hidroponik dengan Sistem NFT

### 2.1.2 Hidroponik *Ebb and Flow* / *Flood and Drain*.

Pada hidroponik *ebb and flow*, tanaman ditanam di dalam sebuah tempat yang berisi media tanam. Media tanam dapat berupa arang sekam ataupun *rock wool*. Nutrisi diberikan secara berseling antara dialirkan dengan tidak dialirkan (dengan batas waktu maksimal tidak dialiri larutan selama 10 menit). Nutrisi ditempatkan dalam sebuah tandon (tempat penampungan), kemudian nutrisi dialirkan ke media tanam. Nutrisi yang telah dialirkan ke media tanam, dimasukkan kembali ke tandon.

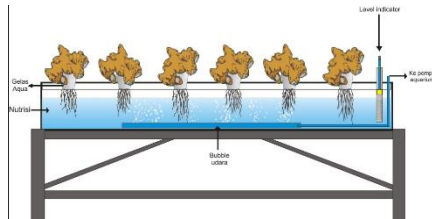


Gambar 2. 2 Hidroponik dengan Sistem *Ebb and Flow*

### 2.1.3 Hidroponik Rakit Ampung (*Floating Raft*)

Pada hidroponik rakit apung, tanaman ditempatkan pada *stereofom* yang diapungkan pada sebuah kolam. Kolam sedalam 40 cm tersebut berisi nutrisi. Pada sistem hidroponik ini perlu ditambahkan *airstone* ataupun *aerator*. *Aerator* berfungsi menghasilkan oksigen untuk pertukaran udara dalam daerah perakaran. Kekurangan oksigen akan mengganggu penyerapan air dan

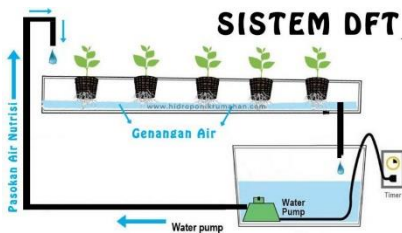
nutrisi oleh akar. Hidroponik rakit apung hanya dapat ditanami oleh tumbuhan yang memiliki bobot rendah.



Gambar 2. 3 Hidroponik dengan Sistem *Floating Raft*

### 2.1.4 Hidroponik *Deep Flow Technique* (DFT)

Hidroponik DFT memiliki prinsip yang hampir mirip dengan NFT, hanya saja pada DFT talang dipasang datar dan larutan hara yang dialirkan memiliki ketinggian 8 cm.



Gambar 2. 4 Hidroponik dengan Sistem *Floating Raft*

## 2.2 *Part Per Million* (PPM) Pada Hidroponik [1]

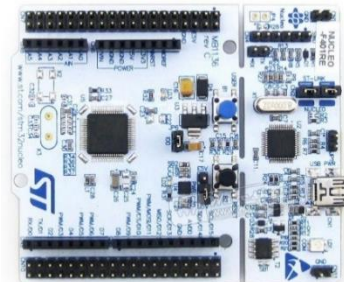
Istilah PPM cukup familiar dikalangan pehobi tanaman hidroponik. PPM yang merupakan singkatan dari “*Part Per Million*” atau “Sepersejuta Bagian” adalah satuan untuk mengukur kepekatan suatu larutan cair. Dalam sistem hidroponik PPM digunakan untuk mengukur tingkat kepekatan larutan nutrisi. Pengukuran kepekatan larutan nutrisi hidroponik diperlukan untuk menyesuaikan kebutuhan nutrisi sesuai dengan fase pertumbuhan tanaman. Penambahan atau peningkatan PPM nutrisi disesuaikan dengan umur tanaman, semakin tua usia tanaman maka semakin tinggi pula PPM yang dibutuhkan.

Tingkat kepekatan nutrisi yang dibutuhkan setiap tanaman berbeda-beda. Misalnya tanaman sayuran buah pada umumnya membutuhkan ppm yang lebih tinggi dibandingkan dengan sayuran

daun. Akan tetapi hal tersebut tidak selalu berlaku, sebab ada juga sayuran daun yang lahap nutrisi dan membutuhkan ppm tinggi. Selain ppm, namun untuk standard pemberian PPM dengan satu sumber pengairan walupun dengan jenis tanaman yang berbeda adalah 500-700 PPM. Kepekatan nutrisi hidroponik diukur dengan sebuah alat yang disebut TDS meter dengan satuan ppm.

### 2.3 STM32 F446RE NUCLEO [2]

STM32 F446RE NUCLEO merupakan mikrokontroler yang diproduksi oleh STMicroelectronics dengan arsitektur *processor* 32-bit RISC (*Reduced instruction Set Computer*). Mikrokontroler STM32 dikembangkan oleh Advanced RISC Machine (ARM) yang dulunya dikenal dengan Acorn RISC Machine. Terdapat beberapa seri *processor* ARM Cortex dari yang memiliki fungsi sederhana hingga yang saat ini banyak digunakan pada *smartphones* mulai dari seri terkecil ARM Cortex-M, ARM Cortex-R hingga ARM Cortex-A. Ada juga seri ARM *Secure Core* dan ARM *Machine Learning* . Arsitektur RISC terdiri dari file *register* besar yang sama, memuat dan menyimpan arsitektur, mode pengalaman sederhana dan bidang instruksi panjang yang sama. Karena karakteristik tersebut ARM mempunyai kinerja tinggi, kode yang simpel, konsumsi daya rendah dan ukuran yang kecil. ARM dapat menjalankan set instruksi yang disebut *Thumb* yang terdiri dari instruksi 32-bit ke 16-bit. Oleh karena itu instruksi 16-bit dan 32-bit dapat digabungkan tanpa mempengaruhi kinerja *processor*.



Gambar 2. 5 Board STM32 F466RE NUCLEO

### 2.4 Sensor TDS [3]

Sensor *Total Dissolved Solid* (TDS) adalah sensor dirancang untuk mengetahui jumlah zat padat yang terlarut dalam air. Hasil



pengukuran TDS Meter mempunyai satuan *Part Per Million* (PPM) yaitu bagian per satu juta. PPM dapat diartikan suatu jumlah ion dalam suatu larutan. Sebagai contoh, terdapat 1 PPM ion Na<sup>+</sup> dalam suatu larutan, ini berarti dalam larutan tersebut terdapat 1 juta ion Na<sup>+</sup>. *Total Dissolved Solid* (TDS) merupakan salah satu indikator tingkat pencemaran air yang sering dianalisis.

Sensor TDS Gravity 1.0 digunakan untuk mengukur kadar TDS (*Total Dissolve Solid*) pada air. TDS sendiri merupakan kadar konsentrasi objek solid yang terlarut dalam air. Semakin tinggi nilai TDS nya maka semakin keruh airnya, begitupun sebaliknya. Semakin rendah nilai TDS nya maka semakin jernih pula air tersebut. Sensor ini mendukung *input* tegangan antara 3,3 - 5V, serta *output* tegangan analog yang dihasilkan berkisar pada 0 - 2,3V. Sangat cocok untuk aplikasi manajemen kualitas air, hidroponik, dsb.

**Spesifikasi:**

- Tegangan *Input*: 3,3 ~ 5,5V
- Tegangan *Output*: 0 ~ 2,3V
- Arus Kerja: 3 ~ 6mA
- Pengukuran TDS: 0 ~ 1000ppm
- Akurasi:  $\pm 10\%$  F.S. (25°C)
- Dimensi Modul: 42 x 32 mm
- Panjang *Probe* 83 cm
- Tipe *Output*: Tegangan Analog



Gambar 2. 6 Probe dan Modul Sensor TDS

## 2.5 Sensor Temperatur DS18B20 [4]

DS18B20 adalah sensor suhu digital yang dikeluarkan oleh *Dallas Semiconductor*. Untuk pembacaan suhu, sensor menggunakan protokol *1 wire communication*. DS18B20 memiliki tiga pin yang

terdiri dari +5V, *Ground* dan data *input/output*. Berikut adalah beberapa fitur DS18B20:

1. Dikalibrasi langsung dalam derajat celsius
2. Jaminan akurasi dari -10°C sampai +85°C
3. Kisaran jangkauan nilai dari -55°C sampai 125°C
4. Tegangan operasi dari 3,0V sampai 5,5V
5. Permintaan waktu kurang dari 750ms



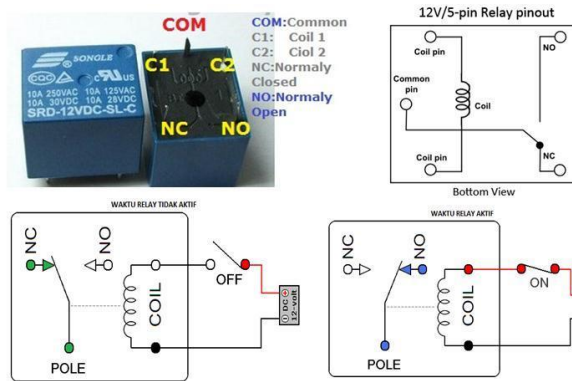
Gambar 2. 7 Bentuk Fisik Sensor DS18B20

## 2.6 Relay [5]

Relay adalah saklar elektro-magnetik yang menggunakan tegangan DC rendah untuk menghidupkan dan mematikan suatu alat atau sistem yang terhubung dengan tegangan DC yang tinggi atau tegangan AC. Susunan relay yang paling sederhana terdiri atas kumparan kawat penghantar yang digulung pada inti besi. Susunan kontak relay, secara umum terdiri dari:

- *Normally Open (NO)* : posisi saklar berada pada keadaan terbuka saat relay dalam keadaan tidak dialiri arus.
- *Normally Close (NC)* : posisi saklar berada pada keadaan tertutup saat relay dalam keadaan tidak dialiri arus.

Berdasarkan pada prinsip dasar cara kerjanya, relay dapat bekerja karena adanya medan magnet yang digunakan untuk menggerakkan saklar. Saat kumparan diberikan tegangan sebesar tegangan kerja relay maka akan timbul medan magnet pada kumparan karena adanya arus yang mengalir pada lilitan kawat. Kumparan yang bersifat sebagai elektromagnet ini kemudian akan menarik saklar dari kontak NC ke kontak NO. Jika tegangan pada kumparan dimatikan maka medan magnet pada kumparan akan hilang sehingga pegas akan menarik saklar ke kontak NC.



Gambar 2. 8 Pin Konfigurasi Relay dan Kondisi NO dan NC Relay

## 2.7 Transistor [5]

Transistor merupakan komponen semikonduktor yang biasanya digunakan sebagai penguat dan sebagai rangkaian sakelar atau penyambung dan pemutus rangkaian. Pada umumnya, transistor memiliki 3 terminal, yaitu *Basis* (B), *Emitter* (E) dan *Collector* (C). Tegangan yang di satu terminalnya misalnya *Emitter* dapat dipakai untuk mengatur arus dan tegangan yang lebih besar daripada arus *input* Basis, yaitu pada keluaran tegangan dan arus *output* Kolektor.

Transistor merupakan komponen yang sangat penting dalam dunia elektronik modern. Dalam rangkaian analog, transistor digunakan dalam *amplifier* (penguat). Rangkaian analog meliputi penguat suara, sumber listrik stabil (*stabilisator*). Dalam rangkaian-rangkaian digital, transistor digunakan sebagai saklar berkecepatan tinggi. Beberapa transistor juga dapat dirangkai sedemikian rupa sehingga berfungsi sebagai *logic gate*, memori dan fungsi rangkaian-rangkaian lainnya.

Telah dipaparkan sebelumnya bahwa salah satu fungsi dari transistor adalah sebagai saklar. Untuk menghasilkan kondisi *on/off* seperti pada saklar, transistor dioperasikan pada salah satu titik kerjanya, titik saturasi dan *cut off*.

Saat kondisi saturasi, transistor seperti sebuah saklar yg tertutup (*on*) sehingga arus dapat mengalir dari kolektor menuju emitor. Sedangkan saat kondisi *cutoff*, transistor seperti sebuah saklar

yg terbuka (*off*) sehingga tidak ada arus yg mengalir dari kolektor ke emitor

$I_C$  adalah arus beban yang akan mengalir dari kaki kolektor ke emitor. Besarnya arus beban ini tidak boleh lebih besar dari  $I_C$  maksimum yang dapat dilewatkan oleh transistor. Arus beban ini dapat dicari dengan persamaan berikut :

Syarat:

$$I_{C(Beban)} < I_{C(Max)}$$

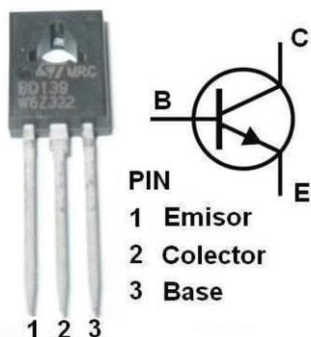
$$I_{C(Beban)} = \frac{V_{CC}}{R_L} \quad (2.1)$$

Setelah arus beban yang akan dilewatkan pada transistor diketahui maka selanjutnya adalah menentukan transistor yg akan dipakai dgn syarat seperti berikut :

$$I_B = \frac{I_{C(Beban)}}{h_{fe}} \quad (2.3)$$

Untuk menentukan  $R_B$ , rumusnya adalah sebagai berikut :

$$R_B = \frac{V_{CC} - V_{BE}}{I_B} \quad (2.4)$$



Gambar 2. 9 Bentuk Fisik, Simbol dan Pin Kofigurasi Transistor

## 2.8 LCD 20 x 4 [6]

*Liquid Crystal Display* (LCD) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama pada setiap rangkaian elektronika saat ini, seperti komputer, kalkulator, dll. Pada kali ini penulis menggunakan LCD seri 20x4, maka pada tampilan yang muncul sebanyak 20 karakter dan 4 baris. Susunan dari titik-titik inilah yang nantinya dapat menampilkan karakter yang beraneka ragam. Pada Tabel 2.1 dijelaskan pin-pin yang terdapat pada LCD 20x4.

Tabel 2. 1 Data Pin LCD 20x4

PIN	Nama	Fungsi
1	Vss	<i>Ground voltage</i>
2	Vcc	+5V
3	Vee	<i>Contras voltage</i>
4	RS	<i>Register select</i> 0 = <i>Instruction Register</i> I = <i>Data Register</i>
5	R/W	<i>Read/Write, to choose write or read mode</i> 0 = <i>write mode</i> I = <i>read mode</i>
6	EN	<i>Enable</i> 0 = <i>start to lacht data to LCD character</i> I = <i>disable</i>
7	DB0	I.SB
8	DB1	-
9	DB2	-
10	DB3	-
11	DB4	-
12	DB5	-
13	DB6	-
14	DB7	MSB
15	BPL	<i>Back Plane Light</i>
16	GND	<i>Ground Voltage</i>

Mengacu pada Tabel 2.1 dapat dijelaskan mengenai EN, RS, RW, yaitu untuk jalur EN dinamakan *Enable*. Jalur ini difungsikan untuk memberitahu LCD bahwa anda sedang mengirimkan sebuah data. Untuk mengirimkan data ke LCD, maka melalui program EN harus dibuat logika *low* “0” dan set pada dua jalur kontrol yang lain RS dan RW. Ketika dua jalur yang lain telah siap, *set* EN dengan logika “1” dan tunggu untuk sejumlah waktu tertentu (sesuai dengan

*datasheet* dari LCD tersebut ) dan berikutnya set EN ke logika *low* “0”.

Kemudian untuk jalur RS adalah jalur *register select*. Ketika RS berlogika *low* “0”, data akan dianggap sebagai sebuah perintah atau instruksi khusus (seperti *clear screen*, posisi kursor, dll ). Ketika RS berlogika *high* “1”, data yang dikirim adalah data *text* yang akan ditampilkan pada *display* LCD. Sebagai contoh, untuk menampilkan huruf “T” pada layar LCD maka RS harus diset logika *high* “1”.

Selanjutnya yang terakhir Jalur RW adalah jalur kontrol *Read/Write*. Ketika RW berlogika *low* (0), maka informasi pada bus data akan dituliskan pada layar LCD. Ketika RW berlogika *high* “1”, maka program akan melakukan pembacaan memori dari LCD. Sedangkan pada aplikasi umum pin RW selalu diberi logika *low* “0”.

Didalam LCD ada beberapa perintah dasar yang harus dipahami, yaitu adalah inisialisasi LCD karakter. Pada Gambar 2.10 ditampilkan gambar bentuk dasar dari LCD ukuran 20x4.



Gambar 2. 10 LCD 20x4

## 2.9 Modul Stepdown LM2596 [7]

Modul *stepdown* lm2596 adalah modul yang memiliki IC LM2596 sebagai komponen utamanya. IC LM2596 adalah sirkuit terpadu / *integrated circuit* yang berfungsi sebagai *step-Down* DC converter dengan *current rating* 3A. Terdapat beberapa varian dari IC seri ini yang dapat dikelompokkan dalam dua kelompok yaitu versi *adjustable* yang tegangan keluarannya dapat diatur, dan versi *fixed voltage output* yang tegangan keluarannya sudah tetap / *fixed*. Gambar 2.11 merupakan bentuk fisik dari modul *stepdown* dengan IC Lm2596.



**Gambar 2. 11**Modul *Stepdown* LM2596

## 2.10 NodeMCU ESP8266 [8]

NodeMCU pada dasarnya adalah pengembangan dari ESP 8266 dengan *firmware* berbasis e-Lua. Pada NodeMcu dilengkapi dengan *micro usb port* yang berfungsi untuk pemrograman maupun *power supply*. Selain itu juga pada NodeMCU di lengkapi dengan tombol *push button* yaitu tombol *reset* dan *flash*. NodeMCU menggunakan bahasa pemrograman Lua yang merupakan *package* dari esp8266. Bahasa Lua memiliki logika dan susunan pemrograman yang sama dengan c hanya berbeda *syntax*. Jika menggunakan bahasa Lua maka dapat menggunakan *tool* Lua *loader* maupun Lua *uploader*.

Selain dengan bahasa Lua NodeMCU juga *support* dengan *software* Arduino IDE dengan melakukan sedikit perubahan *board manager* pada Arduino IDE.

Sebelum digunakan, *board* ini harus di *flash* terlebih dahulu agar *support* terhadap *tool* yang akan digunakan. Jika menggunakan Arduino IDE menggunakan *firmware* yang cocok yaitu *firmware* keluaran dari Ai-Thinker yang *support* AT Command. Untuk penggunaan *tool loader*

**Tabel 2. 2** Spesifikasi NodeMCU

SPESIFIKASI	NODEMCU
Mikrokontroler	ESP8266
Ukuran Board	57 mm x 30mm
Tegangan Input	3,3 ~ 5V

SPESIFIKASI	NODEMCU
GPIO	13 PIN
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Flash Memory	4MB
Clock Speed	40/26/24 MHz
Wifi	IEEE 802.11 b/g/n
Frekuensi	2,4 GHz – 22,5 GHz
USB Port	Micro USB
Card Reader	Tidak Ada
USB to Serial Converter	CH340G



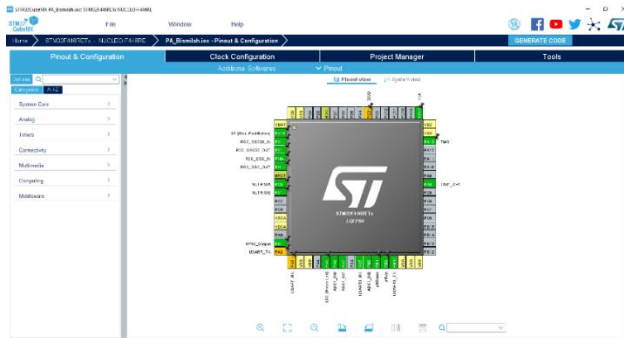
Gambar 2. 12 Board NodeMCU ESP8266

### 2.11 STM32 CubeMX [9]

STM32 CubeMX merupakan alat konfigurasi grafis gratis dengan generator kode tingkat rendah untuk mikrokontroler STM32 ARM Cortex-M. STM32 CubeMX akan memfasilitasi pemilihan Unit mikrokontroler, menyarankan penugasan Pin beserta memeriksa konsistensinya, membuat set instruksi dasar yang nantinya menghasilkan file konfigurasi STM32 untuk *middleware*. STM32 CubeMX merupakan *software* keluaran STMicroelectronics yang difungsikan untuk konfigurasi mikrokontroler berarsitektur ARM serta dapat diunduh secara gratis melalui *web* resminya yaitu [www.st.com/stm32cubemx](http://www.st.com/stm32cubemx). Kemudian dapat diinstal melalui langkah instalasi STM32 CubeMX yang juga disediakan dalam *website*



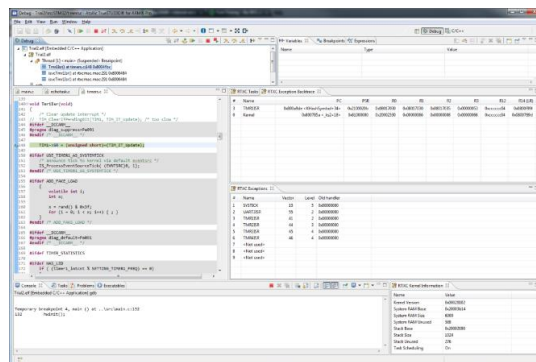
tersebut. STM32 CubeMX dapat dipelajari dengan mudah karena antarmuka konfigurasi mikrokontroler dalam bentuk grafis. Gambar 2.14 Berikut ini merupakan tampilan utama dalam STM32CubeMX :



Gambar 2. 13Tampilan STM32 CubeMX

## 2.12 Atolic TrueSTUDIO [9]

Atolic TrueSTUDIO merupakan *software Integrated Development Environment (IDE)* gratis yang mengintegrasikan *editor* teks untuk menulis program, kompiler yang juga akan mengkonversi kode sumber menjadi file .hex. Atolic TrueSTUDIO akan menyimpan nama file sumber, informasi konfigurasi untuk kompiler, *assembler*, *linker debugger*, *flash loader*, dan utilitas lainnya. Bahasa pemrograman yang digunakan pada Atolic TrueSTUDIO yaitu C/C++.

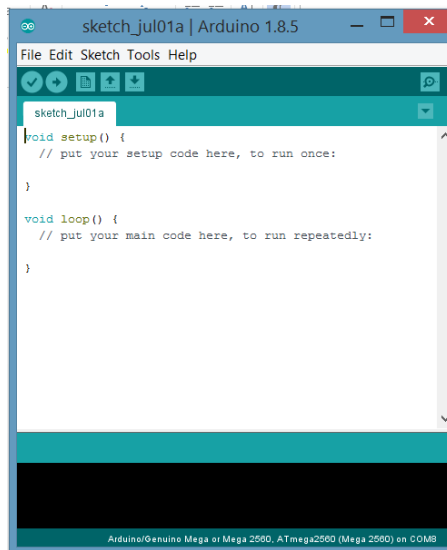


Gambar 2. 14 Tampilan Muka Atolic TrueStudio

### 2.13 Arduino IDE [6]

Arduino IDE adalah perangkat lunak yang digunakan untuk memprogram, monitoring dan *debugging* mikrokontroler Arduino. IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang diberikan melalui sintaks pemrograman.

Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Untuk tampilan dari *sketch Software* Arduino IDE ini dapat dilihat pada Gambar 2.13.

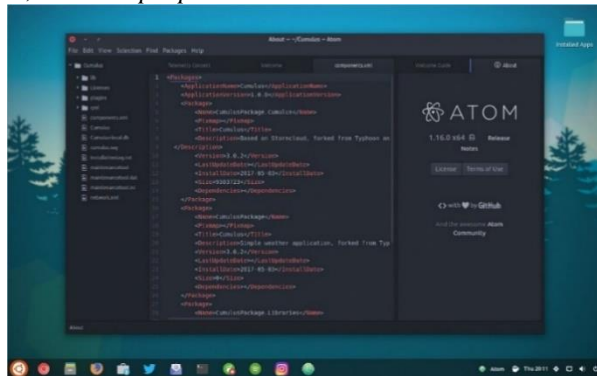


Gambar 2. 15 *Sketch* dari *Software* Arduino IDE

### 2.14 Atom Editor [14]

Atom merupakan *text editor* yang disediakan dengan gratis atau *freeware* yang bertujuan sebagai media untuk membuat sebuah *code* atau program dengan berbagai macam bahasa pemrograman,

seperti: HTML, PHP, dan C++. Atom memiliki beberapa fitur-fitur yang disediakan terhadap *user* untuk dapat memudahkan dalam membangun sebuah sistem, seperti *auto complete text*, *themes*, *cross platform*, dan *multiple panes*.



Gambar 2. 16 Tampilan Atom Editor

-----Halaman ini sengaja dikosongkan-----

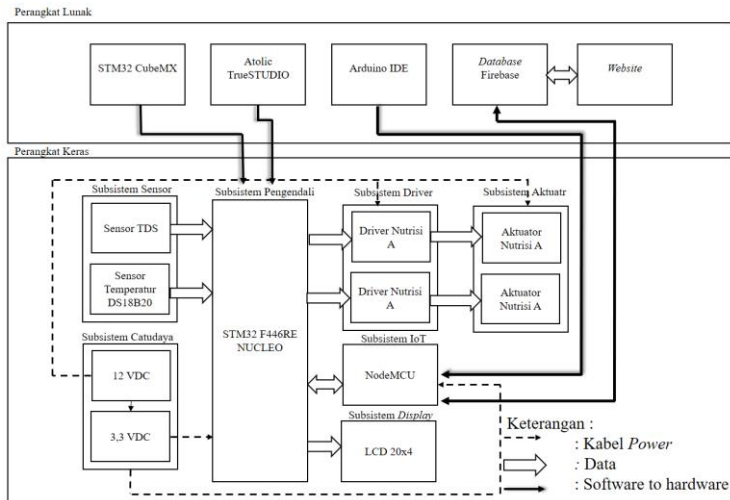
# BAB III

## PERANCANGAN PERANGKAT KERAS DAN PERANGKAT LUNAK

Bab ini membahas perancangan dari perangkat keras dan perangkat lunak sistem informasi dan pengaturan kadar nutrisi serta pembacaan temperatur air untuk pola cocok tanam hidroponik.

### 3.1 Perancangan Sistem Keseluruhan

Perancangan perangkat keras sistem informasi pada kebun hidroponik NFT terbagi menjadi dua bagian penting yaitu perancangan *hardware* dan perancangan perangkat keras.

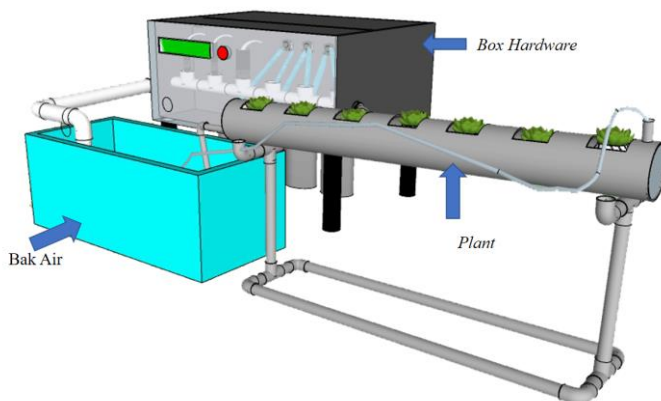


Gambar 3. 1 Diagram Fungsional Sistem

Berdasarkan Gambar 3.1 alur kerja sistem secara umum yaitu, STM32 F446RE sebagai subsistem membaca sensor TDS dan sensor temperatur DS18B20 pada air yang mengalir pada tanaman hidroponik dan hasil pembacaan sensor diolah berdasarkan instruksi yang diberikan oleh Atolic TrueSTUDIO untuk mengaktifkan *driver* sesuai dengan *setpoint* yang telah ditentukan, kemudian *driver* mengaktifkan aktuator yang akan mengalirkan nutrisi A dan nutrisi B

pada air. Selain itu STM32 diberi instruksi untuk menampilkan hasil pembacaan sensor dan kondisi pompa saat aktif pada LCD 20x4 serta mengirim data pembacaan sensor pada NodeMCU. NodeMCU menerima data yang dikirim oleh STM32 sesuai dengan instruksi yang diberikan oleh Arduino IDE kemudian data yang diterima dikirim pada *database* Firebase. Data yang ada pada *database* akan ditampilkan pada *website*. Pada *website* memasukan *setpoint* yang akan tersimpan pada *database*. *Setpoint* tersebut dikirim pada NodeMCU kemudian NodeMCU mengirim data *setpoint* pada STM32 dan STM32 menerima data *setpoint* untuk menentukan aktifnya aktuator.

### 3.2 Perancangan Mekanik



Gambar 3. 2 Desain Mekanik Sistem

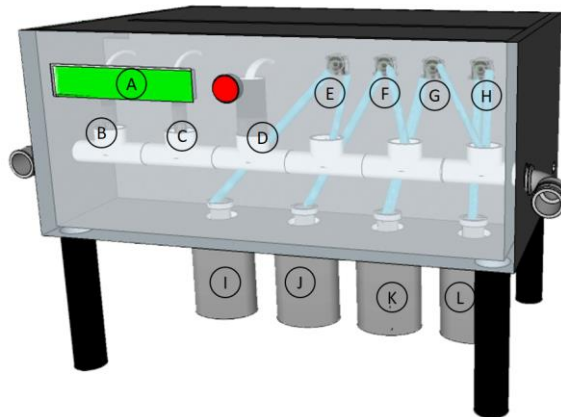
Dalam perancangan mekanik pada Proyek Akhir dilakukan dibagi dua tahap perancangan, perancangan yang dilakukan adalah sebagai berikut:

1. Perancangan *box hardware*
2. Perancangan *plant*

#### 3.2.1 Perancangan *Box Hardware*

*Box hardware* ini berfungsi untuk meletakkan perangkat elektronik yang telah dirancang agar menjadi satu kesatuan sistem.

Perlunya dilakukan perancangan *box hardware* adalah untuk menentukan tata letak setiap subsistem agar dapat bekerja sesuai yang diinginkan, terutama penempatan dan posisi sensor dan aktuator. *Box* ini juga berfungsi sebagai *packaging* sehingga mampu meningkatkan *life time* sistem. Pada Gambar 3.3 Adalah desain *box hardware* yang akan diterapkan pada pengerjaan Proyek Akhir ini.



Gambar 3. 3 Desain *Box Hardware*

Untuk tata letakk komponen pada *box hardware* ditunjukkan pada pada Tabel 3.1

Tabel 3. 1 Letak Komponen Pada *Box Hardware*

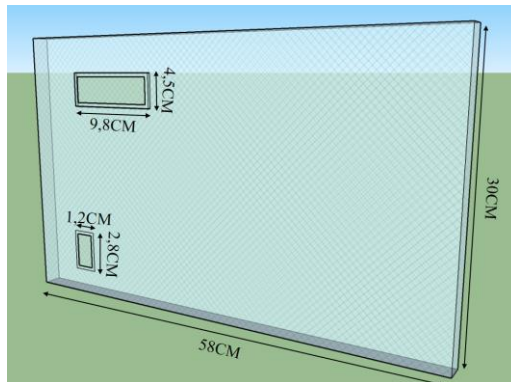
Ket. Gambar	Nama Komponen
A	LCD 20x4
B	Sensor Temperatur DS18B20
C	Sensor TDS
D	Sensor pH
E	Pompa Peristaltik (pH Up)
F	Pompa Peristaltik (pH Down)
G	Pompa Peristaltik (Nutrisi A)
H	Pompa Peristaltik (Nutrisi B)
I	Cairan Ph Up
J	Cairan Ph Down
K	Cairan Nutrisi A

Ket. Gambar	Nama Komponen
L	Cairan Nutrisi B

Untuk spesifikasi *box* yang akan direalisasikan adalah sebagai berikut:

- Material yang digunakan adalah akrilik dengan ketebalan 3mm
- Untuk sisi depan *box* menggunakan akrilik transparan sedangkan sisi lain menggunakan akrilik dengan warna hitam
- Pipa aliran air dan tempat meletakkan sensor menggunakan pipa jenis PVC dengan diameter  $\frac{1}{2}$ ".
- Dimensi setiap sisi *box* ditunjukkan pada Gambar 3.4 – 3

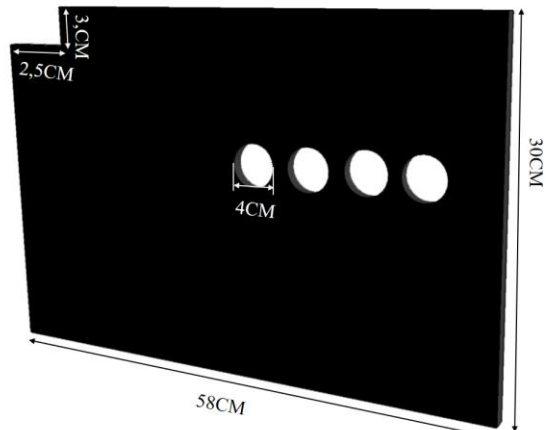
Sisi depan *box* digunakan untuk meletakkan LCD 20x4 yang menampilkan informasi hasil pengukuran dan untuk meletakkan tombol *power* sebagai tombol *on/off* pada alat. Untuk dimensi dan desain sisi depan *box* ditunjukkan pada Gambar 3.4.



Gambar 3. 4 Dimensi Sisi Depan *Box*

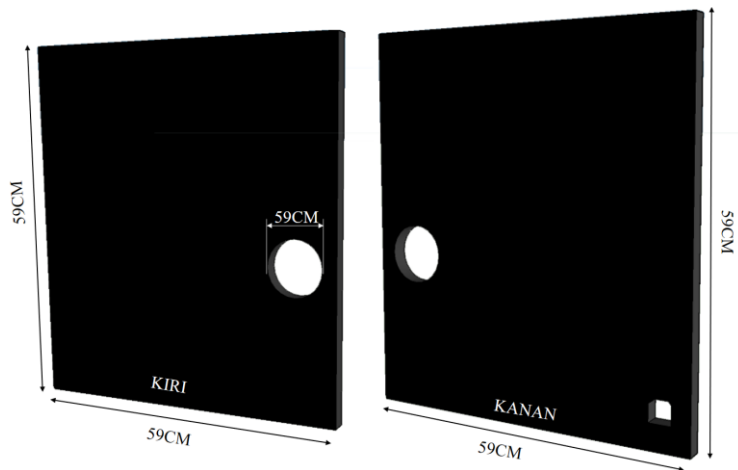
Sisi tengah digunakan untuk meletakkan pompa peristaltik yang mengalirkan cairan nutrisi dari botol menuju pipa penyaluran. Dan sebagai saluran kabel sensor. Untuk dimensi dan desain sisi depan *box* ditunjukkan pada Gambar 3.5.





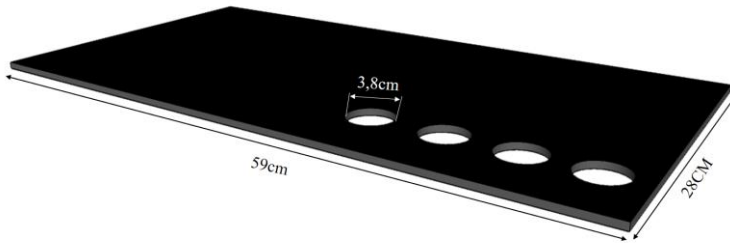
Gambar 3. 5 Dimensi Sisi Tengah *Box*

Sisi samping kanan box digunakan untuk melewati pipa  $\frac{1}{2}$ " yang berfungsi mengalirkan air menuju sensor dan sisi samping digunakan untuk melewati pipa yang mengalirkan air yang telah dibaca sensor dan hasil pengendalian menuju bak air. Untuk dimensi dan desain sisi ditunjukkan pada Gambar 3.6.



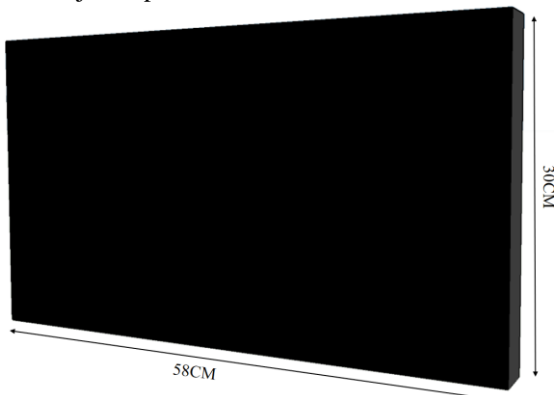
Gambar 3. 6 Dimensi Sisi Samping *Box*

Sisi bawah box digunakan untuk meletakkan botol sebagai wadah cairan nutrisi A dan nutrisi B. Untuk dimensi dan desain sisi bawah *box* ditunjukkan pada Gambar 3.7.



Gambar 3. 7 Dimensi Sisi Bawah *Box*

Sisi bawah box digunakan untuk meletakkan botol sebagai wadah cairan nutrisi A dan nutrisi B. Untuk dimensi dan desain sisi bawah *box* ditunjukkan pada Gambar 3.8.

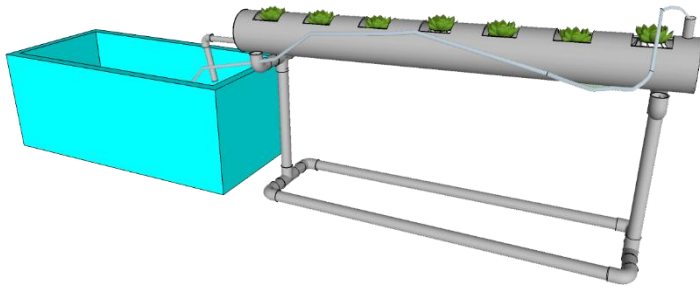


Gambar 3. 8 Dimensi Sisi Bawah *Box*

### 3.2.2 Perancangan *Plant*

Sebelum diaplikasikan langsung pada kebun hidroponik pada perusahaan CV. Panca Gourmet Nusantara perlu dilakukan uji coba alat pada sebuah *plant* yang menyerupai kondisi pada CV. Panca

Gourmet Nusantara. Perancangan *plant* yang akan direalisasikan pada Proyek Akhir ini seperti pada Gambar 3.9.

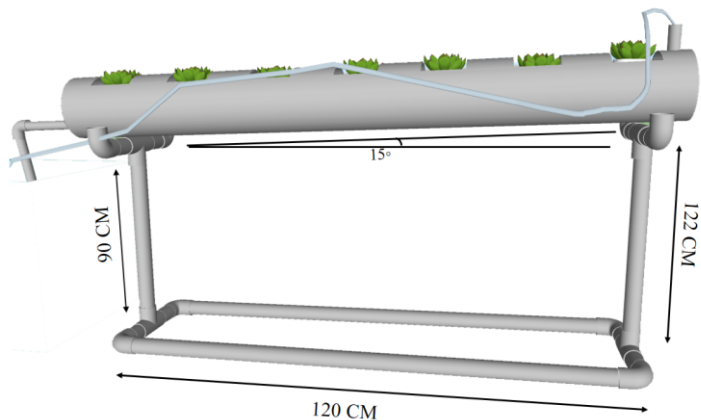


Gambar 3. 9 Desain *Plant* Kebun Hidroponik

Media tanam dibuat menggunakan pipa PVC dengan diameter pipa 2” sedangkan untuk tiang penyangga menggunakan pipa PVC dengan diameter ½” untuk tiang penyangga memiliki tinggi yang berbeda agar wadah tanaman memiliki kemiringan 15 derajat. Dan tanaman yang digunakan untuk pengujian alat ini adalah selada merah. Selain wadah tanaman pompa yang digunakan untuk mengalirkan air menuju tanaman memiliki spesifikasi sebagai berikut:

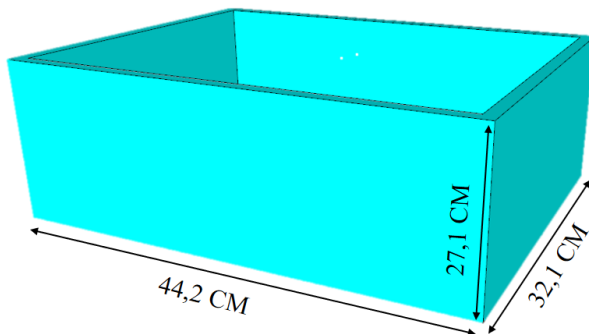
- Pompa celup AA-105
- Sumber tegangan 220 VAC.
- Daya dorong vertikal hingga 3 meter
- Debit pompa 3000 liter/jam

Desain dan ukuran untuk wadah tanaman yang akan direalisasikan ditunjukkan pada Gambar 3.10.



Gambar 3. 10 Dimensi *Plant Media Tanaman*

Untuk bak air yang digunakan menggunakan *box container* yang terbuat dari plastik. Bak air digunakan sebagai wadah penampungan air yang akan disalurkan pada tanaman. Pada Gambar 3.11 merupakan desain penampungan air dan dimensi penampung air yang digunakan.



Gambar 3. 11 Dimensi Wadah Penampungan Air

### 3.3 Perancangan Elektronik

Sedangkan dalam perancangan elektrik pada Proyek Akhir ini terbagi menjadi tujuh buah subsistem. Spesifikasi dari masing masing subsistem tersebut adalah sebagai berikut:

1. Subsistem pengendali  
Untuk pengendalian menggunakan STM32 F446RE NUCLEO yang mengatur segala proses pengolahan data dan pengaturan.
2. Subsistem Sensor
  - a. Sensor TDS untuk membaca jumlah zat terlarut pada air sebagai indikator kadar nutrisi pada air.
  - b. Sensor temperatur untuk membaca temperatur pada air.
3. Subsistem driver  
Menggunakan fungsi relay dan transistor sebagai saklar. Jumlah *driver* sebanyak dua buah, yaitu *driver* untuk aktuator pompa peristaltik.
4. Subsistem aktuator  
Untuk mengalirkan nutrisi A dan B menggunakan pompa peristaltik dengan catu daya 12V dengan kemampuan mengalirkan 100ml cairan selama satu menit.
5. Subsistem catu daya
  - a. 5 VDC untuk subsistem pengendali dan subsistem.
  - b. 12 VDC untuk subsistem aktuator.
6. Subsistem *Display*  
LCD Liquid

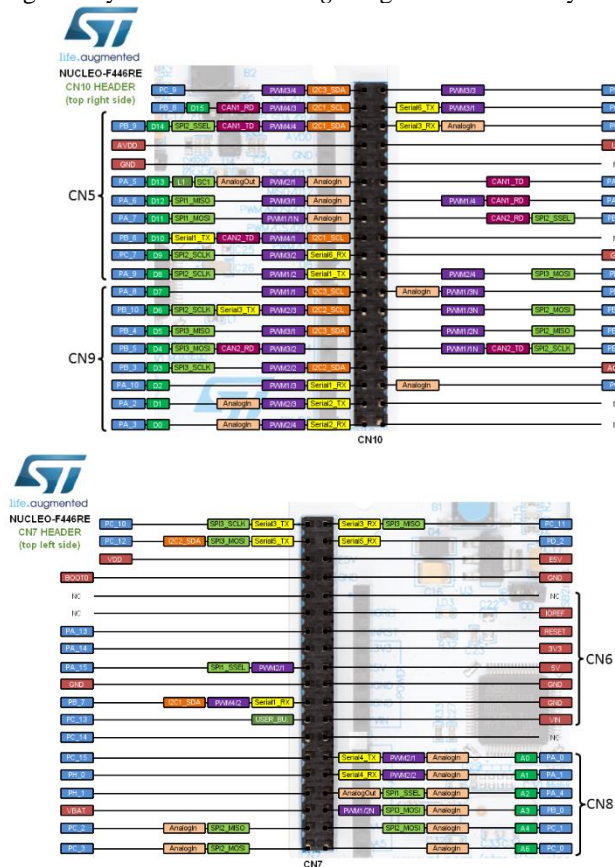
### 3.3.1 Perancangan Subsistem Pengendali

Pada subsistem pengendali menggunakan STM32 F446RE NUCLEO. Subsistem pengendali dirancang agar mampu membaca besaran fisis yang telah dikonversi oleh sensor menjadi besaran listrik serta mampu mengaktifkan aktuator sesuai kondisi yang telah ditentukan. Beberapa fitur yang dimiliki STM32 F446RE NUCLEO sebagai pertimbangan sehingga dipilih sebagai subsistem pengendali adalah

- *floating point unit* dengan frekuensi hingga 180 MHz.
- Tiga 12-bit, 2.4MSPS ADC dengan 16 *channels*
- Dua 12-bit D/A *converters*
- *General-purpose DMA*
- 50 GPIO ports dengan kemampuan *interrupt*
- Empat I2C *interfaces*
- Empat USARTs dan dua UARTs

Selain itu antarmuka yang digunakan STM32 F446RE NUCLEO adalah SWD & JTAG. Sebagai sumber daya pada subsistem pengendali menggunakan subsistem catu daya 5V.

Pada Gambar 3.12 merupakan alokasi dari pin-pin pada STM32 F446RE NUCLEO sehingga kita dapat mengetahui pin yang digunakan sebagai *input* dan *output*, serta mempermudah proses perancangan Proyek Akhir dan *wiring* dengan subsistem lainnya.

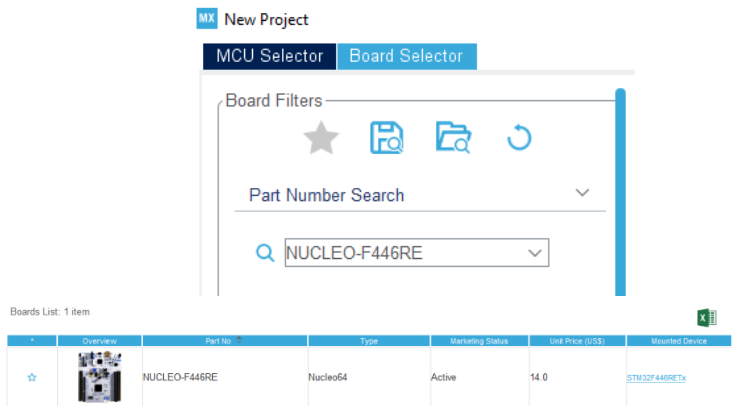


Gambar 3. 12 Alokasi Pin-Pin pada pada STM32 F446RE NUCLEO

Untuk *wiring* subsistem STM32 F446RE NUCLEO dengan subsistem lain atau penggunaan pin-pin pada STM32 F446RE NUCLEO telah dijelaskan pada masing-masing perancangan subsistem yang terhubung dengan subsistem pengendali.

Perancangan pertama yang dilakukan pada subsistem pengendali adalah *setting* konfigurasi pinout dan konfigurasi *clock*. Konfigurasi pinout dilakukan untuk mengatur pin yang akan digunakan sesuai dengan fungsinya. Sedangkan konfigurasi *clock* berfungsi untuk mengatur *clock* internal yang ada dalam mikrokontroler STM32. Sehingga dapat mengatur sumber *clock*, *prescale* dan nilai frekuensi yang diinginkan. Untuk konfigurasi pada STM32 ini menggunakan *software* STM32 CubeMX.

Hal pertama yang dilakukan dalam pengaturan konfigurasi ialah membuat sebuah proyek baru pada STM32 CubeMX kemudian pada tab *board selector* memilih NUCLEO-F446RE.



Gambar 3. 13 Memilih *Board* Pada STM32 CubeMX

Tahap selanjutnya yaitu melakukan konfigurasi pin STM32 yang akan digunakan agar sesuai dengan kegunaannya. Pada Gambar 3.14 Menunjukkan konfigurasi pin yang digunakan pada pengerjaan Proyek Akhir ini.

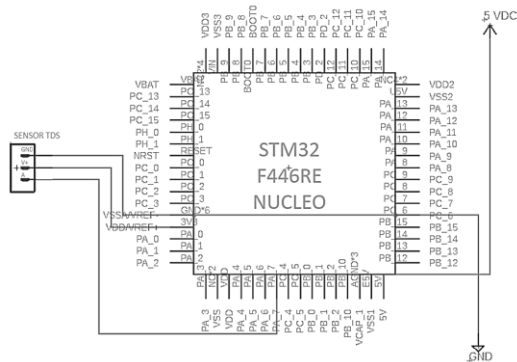




### 3.3.2 Perancangan Subsistem Sensor

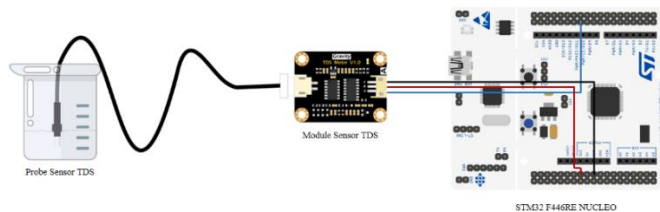
Sesor TDS menggunakan sensor TDS dengan merk Gravity versi V1.0. sensor ini mendukung *input* tegangan 3,3 – 5V. Serta *output* tegangan yang dihasilkan berkisar 0 – 2,3V. Selain itu sensor ini memiliki *range* pengukuran 0 ~ 1000ppm. Yang telah memenuhi kebutuhan pengukuran. Karna dalam pengaturan nutrisi ini dibutuhkan sensor yang mampu membaca hingga 900ppm. Dan sensor ini sangat cocok untuk aplikasi manajemen kualitas air serta hidroponik.

Pada Gambar 3.16 adalah perancangan *wiring* sensor TDS dengan subsitem pengendali.



Gambar 3. 16 Wiring Diagram DS18B20 dengan STM32 F446RE

Berdasarkan diagram listrik yang terdapat pada Gambar 3.16 Untuk *wiring* antara STM32 dengan LCD menggunakan bantuan *software fritzing* ditampilkan pada Gambar 3.17.



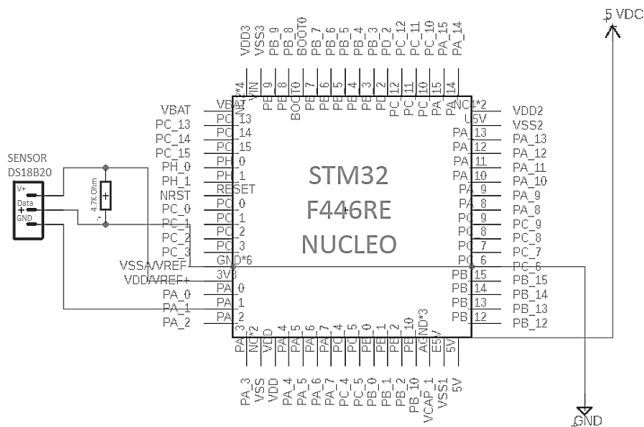
Gambar 3. 17 Wiring Diagram Sensor TDS dengan STM32 F446RE NUCLEO

Berdasarkan Gambar 3.17 pin konfigurasi antartara sensor TDS dengan STM32 F446RE akan ditunjukan pada Tabel 3.2 dan untuk alokasi pin pada STM32 F446RE telah ditunjukan pada Gambar 3.2.

Tabel 3. 2 Pin Konfigurasi Sensor TDS dengan STM32 F446RE NUCLEO

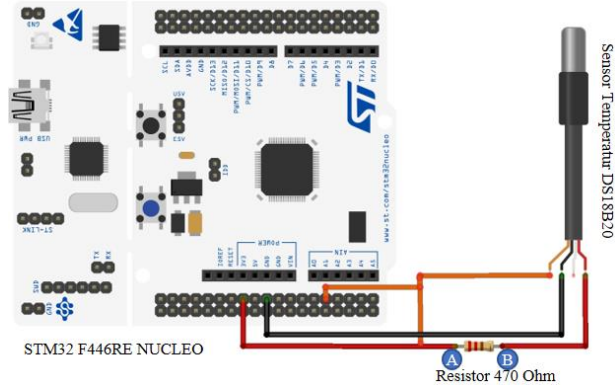
STM32 F446RE NUCLEO	Sensor TDS
PIN 3V3	PIN (+)
PIN GND	PIN (-)
PIN PA7	PIN (A)

DS18B20 merupakan sensor suhu digital yang mampu membaca temperatur dengan ketelitian 9 hingga 12-bit, dengan rentang  $-55^{\circ}\text{C}$  hingga  $125^{\circ}\text{C}$  dengan ketelitian  $(\pm 0,5^{\circ}\text{C})$ . Antarmuka hanya menggunakan satu kabel sebagai komunikasi (menggunakan protokol *1-wire*). Juga mampu diumpankan daya melalui jalur daanya. Rentang dayanya adalah 3,0V hingga 5,5V. Bisa mengkonversi data suhu ke 12-bit *digital word* hanya dalam 750 milidetik. Dan diproduksi dengan *casing* yang tahan air sehingga cocok digunakan untuk mengukur temperatur pada air. Rangkaian listrik antara sensor DS18B20 dengan STM32 ditampilkan pada Gambar 3.18.



Gambar 3. 18 Wiring Diagram DS18B20 dengan STM32 F446RE

Berdasarkan diagram listrik yang terdapat pada Gambar 3.18 Untuk *wiring* antara STM32 dengan LCD menggunakan bantuan *software fritzing* ditampilkan pada Gambar 3.19.



Gambar 3. 19 Diagram Koneksi Sensor Temperatur DS18B20 dengan STM32 F446RE NUCLEO

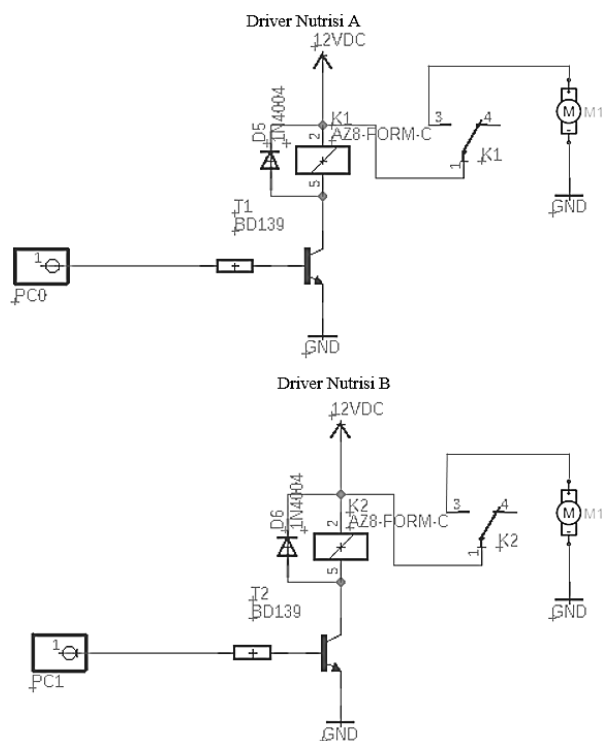
Berdasarkan Gambar 3.9 pin konfigurasi antartara sensor TDS dengan STM32 F446RE akan ditunjukan pada Tabel 3.3 dan untuk alokasi pin pada STM32 F446RE telah ditunjukan pada Gambar 3.2.

Tabel 3. 3 Pin Konfigurasi Sensor Temperatur DS18B20 dengan STM32 F446RE NUCLEO

STM32 F446RE NUCLEO	Sensor DS18B20	Resisto 470 Ohm
PIN 3V3	PIN(+)	A
PIN GND	PIN (-)	-
PIN PA1	PIN (A)	A
-	PIN (+)	B

### 3.3.3 Perancangan Subsistem *Driver*

*Driver* pada Gambar 3.20 digunakan untuk menggerakkan aktuator sesuai program yang telah dibuat. *Driver* ini terdiri dari komponen resistor, dioda, transistor dan relay.



Gambar 3. 20 Skematik Driver Aktuator Pompa Peristaltik

Berikut adalah spesifikasi komponen yang dibutuhkan untuk membuat *driver* dicantumkan pada Tabel 3.4.

Tabel 3. 4 Komponen Pembuatan *Driver*

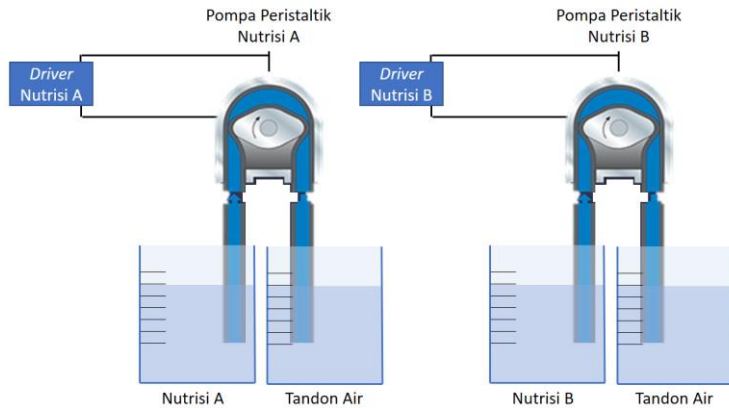
No.	Komponen	Spesifikasi
1	Resistor	1000hm
2	transistor	BD139
3	Dioda	1N4002
4	relay	HLS8L-DC12V-S-C

Untuk perancangan *driver* memanfaatkan fungsi transistor sebagai saklar. Untuk dapat bekerja seperti saklar *off* transistor bekerja pada

kondisi *cutoff* sedangkan untuk dapat bekerja seperti saklar *on* transistor bekerja pada kondisi saturasi.

### 3.3.4 Perancangan Subsisitem Aktuator

Aktuator yang digunakan adalah pompa peristaltik 12V. Aktuator berfungsi untuk mengalirkan nutrisi A dan B kedalam tandon air hidroponik. Pada perancangan aktuator dibuat desain letak pompa dan letak botol pada sistem. Sehingga aktuator mampu mengalirkan cairan nutrisi dengan tepat.

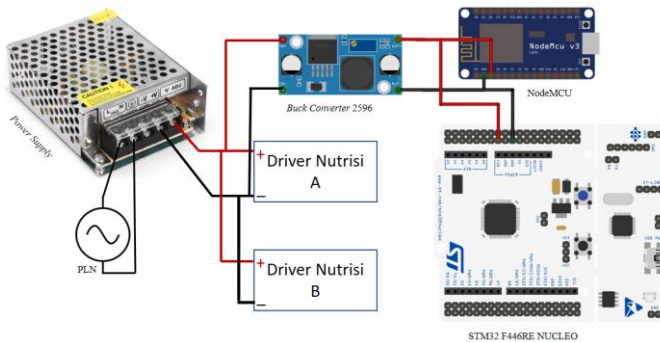


Gambar 3. 21 Koneksi Rangkaian *Driver* Dengan Pompa Peristaltik

### 3.3.5 Perancangan Subsisitem Catudaya

Rangkaian catudaya sangat dibutuhkan untuk bekerjanya sistem, karena semua sistem tergantung dari supply energi. Dalam subsistem catu daya, sumber tegangan sistem berasal dari jala-jala PLN 220 VAC.

Untuk mendapatkan tegangan sebesar 12 VDC digunakan *power supply switching* 12 VDC 3A. Sedangkan untuk mendapatkan tegangan sebesar 5 VDC digunakan *buck converter* dengan IC LM2596S, *buck converter* berfungsi untuk menurunkan tegangan 12 VDC dari *power supply* menjadi 5 VDC. Tegangan 5 VDC ini berfungsi sebagai sumber tegangan untuk subsistem pengendali dan subsistem *server*. Pada Gambar 3.22 Menunjukkan rancangan skematik untuk perancangan subsistem catu daya.



Gambar 3. 22 Rangkaian Catu Daya dengan Subsisitem lain

### 3.3.6 Perancangan Subsisitem *Display*

LCD digunakan untuk menampilkan data yang diberikan oleh STM32 melalui program yang dibuat. LCD akan menampilkan segala bentuk kerja yang sedang dilakukan oleh STM32 yaitu hasil pengukuran sensor dan kondisi aktuator saat menyala. Gambar 3. Menunjukkan gambar konfigurasi pin LCD 16 karakter x 4 baris.

Modul LCD dikendalikan secara parallel baik untuk jalur data maupun kontrolnya shingga membutuhkan 6 atau 7 pin. Dengan demikian untuk kontroler yang sibuk dan harus mengendalikan banyak I/O, menggunakan jalur parallel adalah solusi yang kurang tepat. Maka digunakanan I2C adapter, yaitu modul LCD yang dikendalikan secara serial sinkron dengan protokol I2C/IIC (*Inter Integrated Circuit*). Gambar 3.23 Merupakan gambar rangkaian LCD dihubungkan dengan STM32 F446RE NUCLEO.



untuk alokasi pin pada STM32 F446RE telah ditunjukkan pada Gambar 3.24.

Tabel 3. 5 Pin Konfigurasi Sensor Temperatur DS18B20 dengan STM32 F446RE NUCLEO

STM32 F446RE NUCLEO	I2C Adaptor
PIN 3V3	PIN VCC
PIN GND	PIN (GND)
PIN PA3	PIN (SDA)
PIN PB12	PIN (SCL)

### 3.3.7 Perancangan Subsistem *Internet of Thing (IoT)*

Untuk platform IoT yang digunakan adalah NodeMCU. NodeMCU merupakan perangkat keras berupa *System On Chip* ESP8266. Untuk pemrogramannya menggunakan Lua Script. Spesifikasi yang dimiliki NodeMCU yang menjadi aspek dipilihnya NodeMCU sebagai subsistem IoT adalah sebagai berikut:

- *Wireless standard*: IEEE 802.11 b/g/n
- 6 x Digital I/O, 3 x PWM *channels*, 1 x ADC *channel*
- Semua I/O dapat dikendalikan melalui jaringan WiFi
- GPIO mampu mengalirkan arus sebesar 15 mA
- Mendukung *Smart Link intelligent networking*
- Built-in 32-bit MCU
- Built-in TCP/IP *protocol stack* dan mendukung *multiple TCP Client connection*
- UART/GPIO *data communication interface*

NodeMCU sebagai subsistem IoT berfungsi untuk menerima data hasil pengukuran masing-masing sensor dan kondisi aktuator yang diperoleh dari STM32 F446RE NUCLEO melalui komunikasi serial. Yang kemudian data-data tersebut akan dikirim pada *database*. Serta NodeMCU juga menerima data *set-point* dari *database*, yang kemudian juga akan dikirimkan pada STM32 F446RE NUCLEO. Dan untuk diagram wiring NodeMCU dengan STM32 F446RE NUCLEO ditunjukkan pada Gambar 3.25.



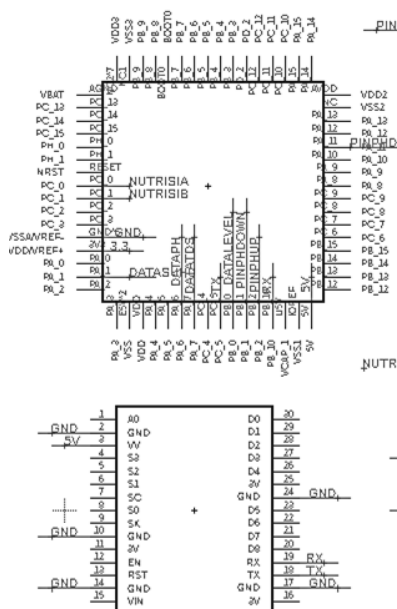


Tabel 3. 6 Pin Konfigurasi Sensor TDS dengan STM32 F446RE NUCLEO

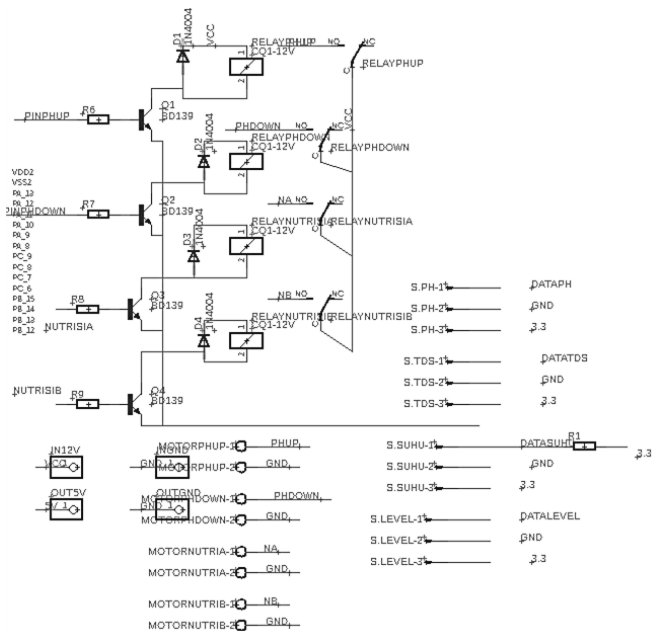
STM32 F446RE NUCLEO	NodeMCU
PIN PA3 (Rx)	PIN Rx
PIN PA2 (Tx)	PIN Tx
PIN GND	PIN GND

### 3.3.8 Perancangan *Board* PCB

Perancangan *board* PCB bertujuan untuk memudahkan dalam proses perangkaian komponen yang dibutuhkan sistem. Sehingga perancangan sistem dapat ter realisasikan. Untuk pembuatan desain PCB dikerjakan menggunakan *software Eagle Pro*. Desain PCB dibuat sesuai dengan perancangan setiap subsistem. Gambar 3.27 menunjukan rangkaian skematik untuk pada *board* PCB.



Gambar 3. 27 Skematik PCB Pengendali Menggunakan *Software Eagle Pro*



Gambar 3. 28 Skematik PCB Sensor dan *Driver* menggunakan *software Eagle Pro*

Setelah desain rangkaian selesai dibuat selanjutnya dilakukan pencetakan. Setelah dilakukan pencetakan dilakukan pengecekan apakah terjadi hubungan singkat antar komponen yang seharusnya tidak terhubung. Pengecekan dilakukan menggunakan multimeter.

### 3.4 Perancangan *Software*

Pada tahap ini dilakukan sebuah perancangan agar sistem yang dibuat mampu bekerja sesuai dengan instruksi yang diberikan. Pada tahap ini dilakukan perancangan dengan menggunakan beberapa *software* yang dibutuhkan.

Untuk perancangan *software* pada pengerjaan Proyek Akhir ini dilakukan dua tahap, tahapan tersebut adalah:

1. Perancangan program STM32 F446RE.
2. Perancangan proram NodeMCU
3. Perancangan *website* sebagai sistem informasi.

### 3.4.1 Perancangan Program STM32F446RE

Pemrograman ini dilakukan untuk mengatur kinerja STM32 F446RE NUCLEO, dimana mikrokontroler STM32 F446RE NUCLEO merupakan otak subsistem pengendali. Perangkat lunak pada mikrokontroler berfungsi untuk melakukan pengendalian seluruh subsistem. Selain dilakukan pemrograman pada STM32 F446RE juga dilakukan pemrograman pada NodeMCU yang merupakan otak pada subsistem IoT.

Spesifikasi *software* yang digunakan pada perancangan program mikrokontroler terdapat pada tabel 3. .

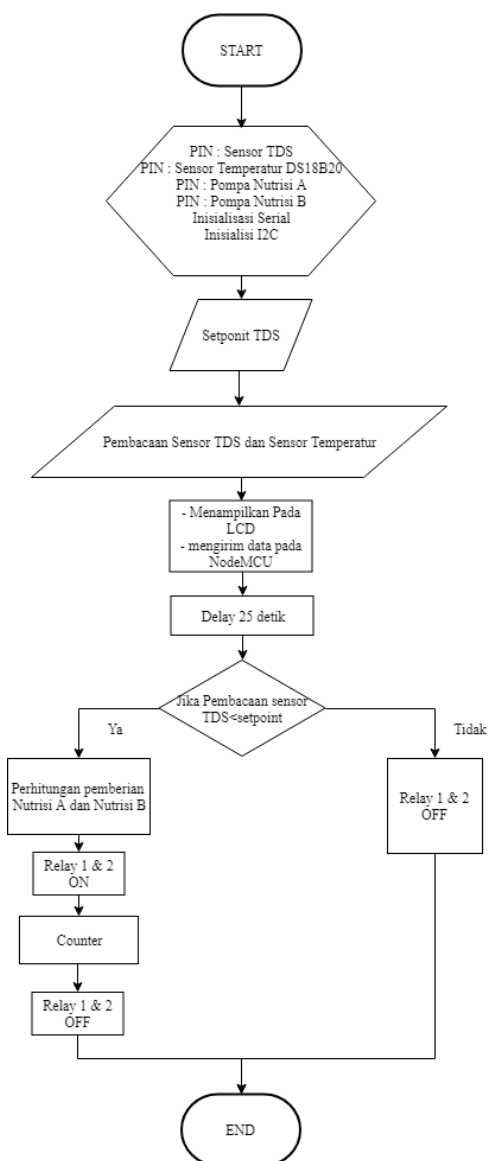
Tabel 3. 7 Tabel *Software* yang Digunakan

No.	<i>Software</i>	Fungsi
1.	STM32 CubeMX	konfigurasi STM32 F446RE NUCLEO dengan generator kode
2.	Atolic TrueSTUDIO	Perogram editor dan sebagai kompailer untuk STM32 F446RE NUCLEO, bahasa pemrograman yang digunakan yaitu C/C++

Algoritma perangkat lunak dibuat untuk mempermudah pembuatan program dari keseluruhan sistem. Gambar 3.29 Merupakan diagram alir secara umum sistem.

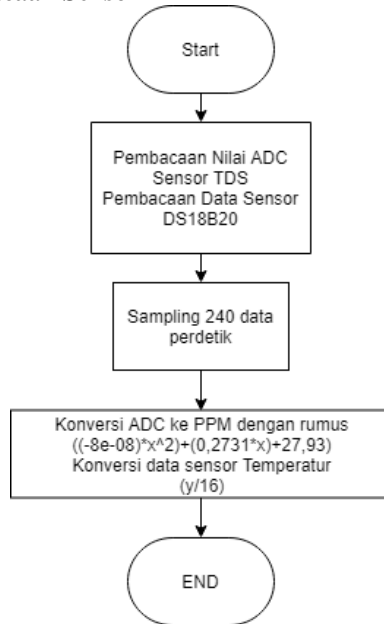
Berdasarkan diagram alir pada Gambar 3.29 untuk melakukan perancangan program ini dapat dikelompokkan menjadi lima tahap perancangan, berikut adalah lima tahap tersebut.

1. Pengaturan konfigurasi pada STM32 F446RE NUCLEO
2. Pembacaan Sensor
3. Pengendalian Aktuator
4. Menampilkan data Pada LCD
5. Serial Komunikasi



Gambar 3. 29 Diagram Alir Sistem

#### 3.4.1.1 Pembacaan Sensor



Gambar 3. 30 Flowchart Pembacaan Sensor

Terdapat dua sensor yang akan dibaca hasil pengukurannya, yaitu sensor TDS dan sensor temperatur DS18B20. Untuk dapat membaca hasil pengukuran sensor perlu dilakukan pemrograman pada STM32 F446RE NUCLEO. Pemrograman dilakukan menggunakan *software* Atollic TrueSTUDIO. Untuk bahasa pemrograman yang digunakan adalah bahasa C.

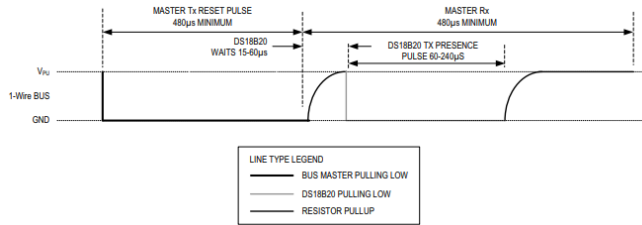
Untuk pembacaan sensor TDS dilakukan pemrograman *Analog Digital Converter*(ADC). Pembacaan ADC dilakukan menggunakan metode *Direct Memory Access* (DMA). Metode DMA dipilih karena dengan menggunakan metode DMA tidak mempengaruhi program yang lain.

Dari hasil pembacaan ADC dilakukan sampling sebanyak 250 data setiap detiknya.

Dari hasil sampling data ADC dilakukan konversi pada satuan ppm, konversi dilakukan menggunakan persamaan yang diperoleh dari hasil kalibrasi.

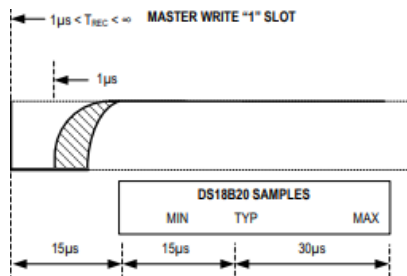
Sedangkan untuk pembacaan hasil pengukuran sensor temperatur DS18B20 harus menyesuaikan dengan *data sheet* sensor karena sensor DS18B20 merupakan sensor digital, sehingga program yang dibuat harus sesuai dengan spesifikasi sensor.

Yang harus dilakukan untuk pembacaan sensor DS18B20 yaitu selalu melakukan inisialisasi pada sensor. Dengan cara memberi data *low* selama 480us sesuai *data sheet* yang tertera pada Gambar 3.31



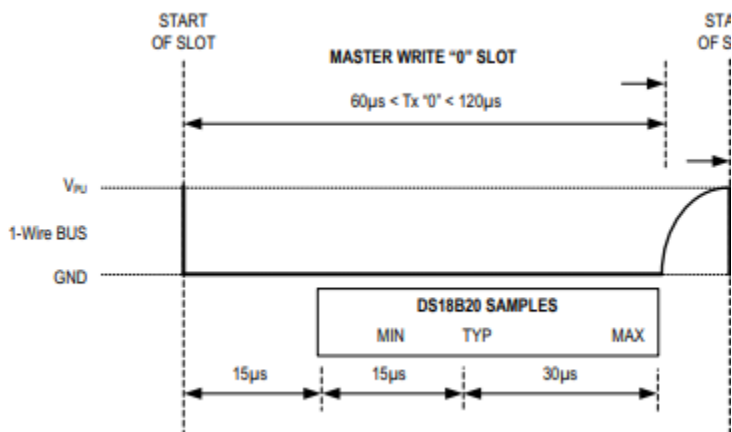
Gambar 3. 31 Inisialisasi Sensor DS18B20

Untuk menuliskan bit ke sensor perlu dilakukan beberapa operasi pada jalur data. Untuk menghasilkan data 1. Setelah *pulling* 1-*Wire bus* ke *low*. Master harus melepaskan 1-*Wire* dalam waktu 15us. Saat bus *release*. Resistor *pullup* 5KOhm akan *pull bus* ke *high*. Penulisan bit 1time slot ini dijelaskan pada *data sheet* yang ada pada Gambar 3.31.



Gambar 3. 32 Master Write "1" Slot

Untuk menghasilkan *time slot write* “0”, setelah memberi logika *low* pada bus 1-wire, master harus terus menahan bus agar tetap berlogika *low* selama durasi slot waktu (setidaknya 60µs). Keterangan Master write “0” slot. Penulisan bit 0 time slot ini dijelaskan pada *data sheet* yang ada pada Gambar 3.33.

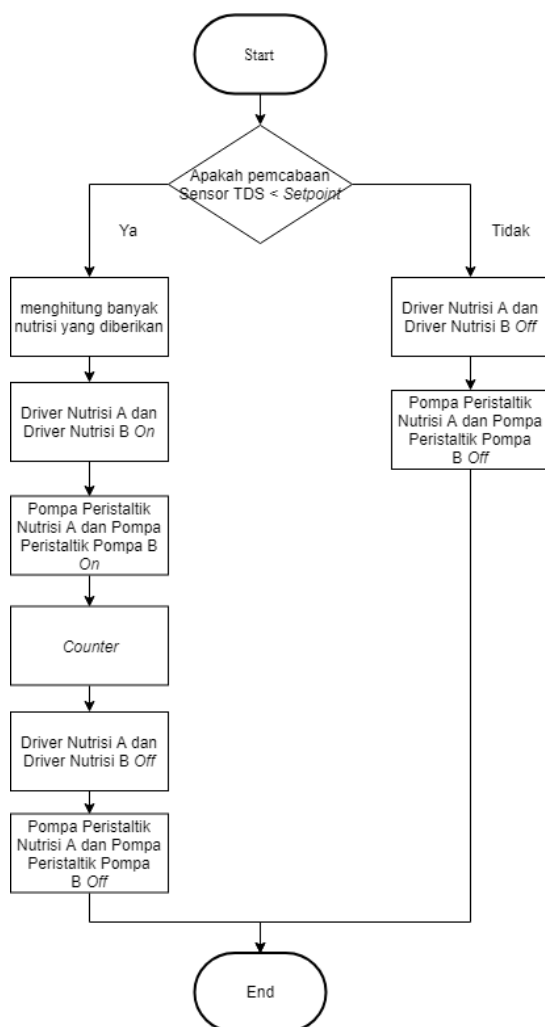


Gambar 3. 33 Master Write “0” Slot

#### 3.4.1.2 Pengendalian Aktuator

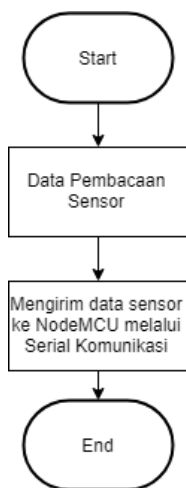
Pengendalian aktuator diprogram agar mampu mengatur kadar nutrisi sesuai *setpoint*. Yaitu dengan mengaktifkan aktuator jika nilai TDS tidak sesuai dengan *setpoint* yang ditetapkan. Dan dilakukan pemrograman untuk mengatur lama waktu aktuator aktif. Setelah dilakukan pengendalian kadar nutrisi jika hasil pengendalian masih kurang dari *setpoint* pengendalian akan dilakukan kembali setelah selang beberapa menit, hal ini dilakukan untuk memastikan bahwa cairan yang diberikan telah tercampur secara keseluruhan pada air.



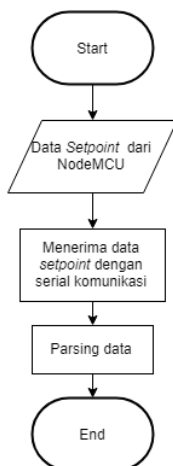


Gambar 3. 34 *Flowchart* Pengendalian Aktuator

### 3.4.1.3 Serial Komunikasi Dengan NodeMCU



Gambar 3. 35 *Flowchart* Pengiriman Data STM32 F446RE ke NodeMCU



Gambar 3. 36 *Flowchart* Penerimaan Data STM32 F446RE Oleh NodeMCU

Serial komunikasi dilakukan untuk mengirim data hasil pengukuran untuk setiap sensor dan kondisi pompa apakah pada kondisi aktif atau sedang tidak aktif pada STM32 F446RE NUCLEO.

Selain mengirimkan data hasil pembacaan sensor kepada NodeMCU. STM32F446RE NUCLEO juga menerima data *setpoint* dari NodeMCU. Untuk menerima data dari NodeMCU menggunakan serial komunikasi dengan fungsi *interrupt*. Fungsi *interrupt* dipilih karena dalam penerimaan data dilakukan secara terus menerus dan tidak terjadi hambatan karena adanya program lain. Serta tidak menghalangi berjalannya program lain.

### **3.4.2 Pemrograman NodeMCU**

NodeMCU sebagai subsistem IoT berperan penting dalam melakukan pengolahan data untuk pengiriman data hasil pembacaan sensor pada *database*. Dan pengiriman data *setpoint* yang terdapat pada *database* yang digunakan untuk mengatur kadar pH dan TDS.

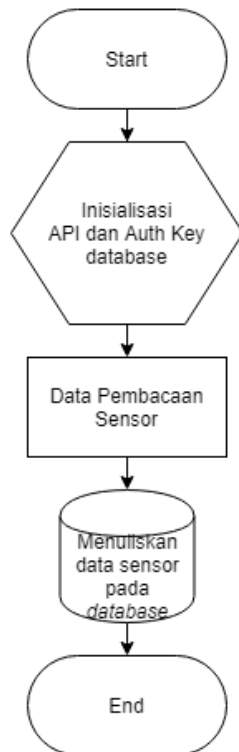
Dalam pemrograman NodeMCU menggunakan *software* Arduino IDE. Terdapat empat proses yang dilakukan oleh NodeMCU yaitu menerima data pembacaan sensor dari STM32 F446RE dan mengirimkan data pembacaan sensor pada *database* dan menerima data *setpoint* dari *database* lalu mengirimkan data *setpoint* pada STM32F446RE.

#### **3.4.2.1. Serial Komunikasi Dengan STM32 F446RE**

NodeMCU menerima data pembacaan sensor oleh STM32 F446RE menggunakan serial komunikasi, data yang diterima oleh NodeMCU kemudian dilakukan penguraian data sesuai dengan tipe datanya. Penguraian data dilakukan untuk memudahkan pengolahan data dan pengiriman pada *database*.

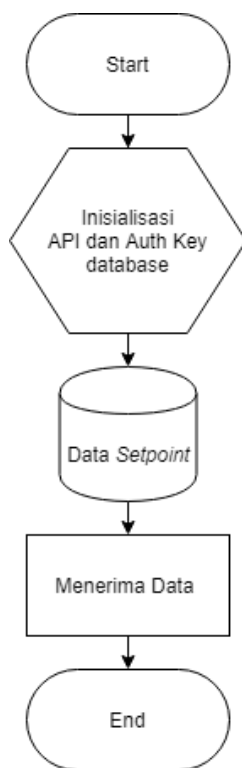
Data *setpoint* yang telah diterima oleh NodeMCU dari *database* dikirimkan ke STM32 F446RE melalui serial komunikasi data *setpoint* tersebut yang menentukan kapan dilakukan pengaturan pH dan nutrisi pada sistem.

#### 3.4.2.2. Pengiriman dan Penerimaan Data Pada *Database*



Gambar 3. 37 *Flowchart* Pengiriman Data Pada *Database* Melalui NodeMCU

Untuk dapat mengirim data pembacaan sensor pada *database* terlebih dahulu mengubah *rule* yang ada pada *database* untuk *write* menjadi *true*. Kemudian harus menginisialisasi API dan *auth key* yang didapatkan pada proyek yang dibuat pada firebase. Kemudian mengirim data menggunakan *library* firebase dengan fungsi *firebase.setFloat()* dan *firebase.setInt()*.



Gambar 3. 38 *Flowchart* Pembacaan Data Pada *Database* Oleh NodeMCU

Sedangkan untuk mendapatkan data *setpoint* pada *database* perlu mengubah rule pada *database* untuk *read* menjadi *true*. Kemudian menggunakan fungsi `firebase.getFloat()` yaitu fungsi dari *library* *firebase*.

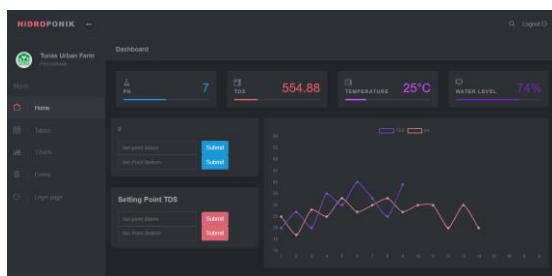
### 3.4.3 Perancangan *Website* Sebagai Sistem Informasi

Pembuatan *flowchart* dilakukan untuk menentukan algoritma dari perancangan *website*. berdasarkan *flowchart* yang dibuat, alur kerja *website* ialah.

Terdapat *login page* dimana *user* harus memasukan terlebih dahulu *username* dan *password* yang telah terdaftar. Jika *username* dan *password* yang dimasukan tidak sesuai maka *user* tidak dialihkan ke halaman utama dan terdapat peringatan. Sedangkan jika *username* dan *password* yang dimasukan dikenali atau sesuai dengan yang terdaftar maka *user* dialihkan ke halaman muka (*index.html*). pada halaman muka

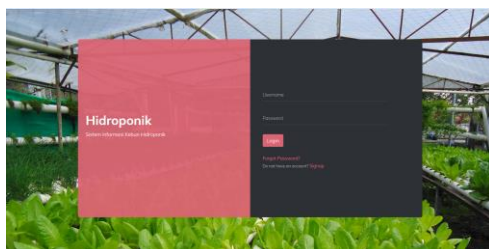
### 3.4.3.1 Perancangan Tampilan Website

Tampilan pada *website* dirancang agar memiliki tampilan yang menarik dan menampilkan data sesuai yang diinginkan. Perancangan tampilan *website* dilakukan menggunakan *software* Atom Editor. Menggunakan bahasa *html*, *java script* dan *css*. Tampilan pada *website* yang diinginkan adalah seperti yang tertera pada Gambar 3.39 .



Gambar 3. 39 Desain Tampilan Website

Selain tampilan halaman muka *website* juga dilakukan desain untuk *login page*. Gambar 3.40 Merupakan desain yang akan dibuat untuk tampilan *login page*.



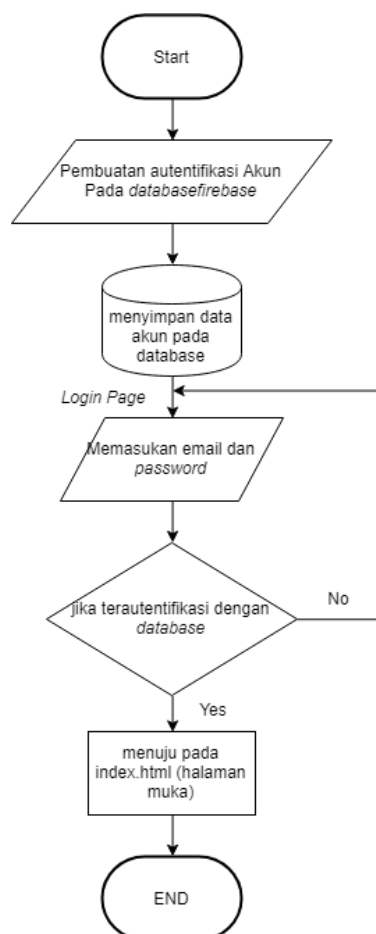
Gambar 3. 40 Desain Tampilan Login Page

setelah dilakukan desain tampilan kemudian dilakukan pemrograman *website* agar sesuai dengan desain yang diberikan. Pemrograman tampilan *website* dikerjakan menggunakan bahasa *Hypertext Markup Language* (HTML) untuk mengisi konten dan informasi apa saja yang ditampilkan pada *website*. Sedangkan untuk menyesuaikan tampilan *website* dengan desain yang telah dibuat menggunakan *Cascading Style Sheets* (CSS).

#### **3.4.3.2 Perancangan Login Page**

Perancangan *login page* dilakukan dengan memilih metode *login*, metode login yang dipilih adalah dengan email dan kata sandi. Kemudian membuat sebuah akun yang terautentifikasi dengan *database*. Dapat menggunakan *Firebase Authentication* untuk mengizinkan pengguna melakukan autentikasi dengan *Firebase* menggunakan alamat email dan sandi yang telah dibuat.

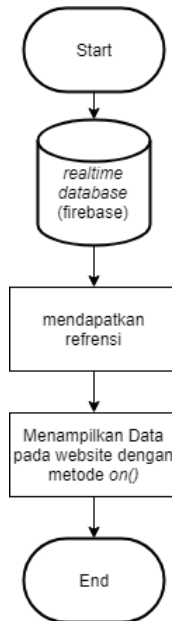
Pengguna memasukan email dan sandi pada *login page* program dibuat dengan memasukan program berikut “*signInWithEmailAndPassword*”. Jika akun email dan sandi terautentifikasi oleh *firebase* maka menuju ke *index.html* yaitu halaman muka *website*. Jika akun yang dimasukan tidak terautentifikasi maka dilakukan *handle error* yaitu terdapat peringatan “email atau kata sandi yang dimasukan salah ” dan tetap berada pada *login page*. Perancangan algoritma untuk *login page* sesuai dengan *flowchart* yang terlampir pada Gambar 3.40.



Gambar 3. 41 *Flowchart Proses Pada Login*



### 3.4.3.3 Menampilkan Data Dari *Database* Pada *Website*



Gambar 3. 42 *Flowchart* Pembacaan Data Pada *Database*

*Database* yang digunakan adalah firebase sedangkan struktur *database* pada firebase adalah NoSQL. Untuk mampu membaca *database* pada firebase perlu dilakukan inisialisasi untuk mengonfigurasi Firebase JavaScript SDK untuk menggunakan *Authentication*, *cloud storage*, *realtime database* dan *cloud firestore*.

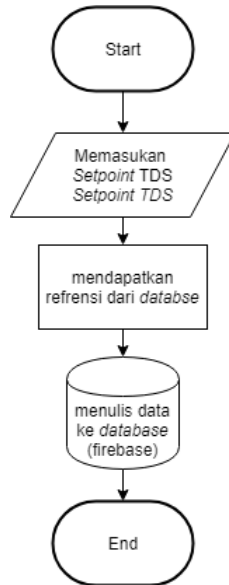
Setelah dilakukan inisialisasi maka untuk menampilkan data pada *website* yang ada di *database* perlu dilakukan pemrograman menggunakan JavaScript. Pemrograman dilakukan menggunakan *software* Atom Editor.

Untuk dapat membaca data pada *database* harus mendapat refrensi dari *database*.

Data Firebase diambil dengan menambahkan *listener* asinkron ke *firebase.database.Reference*. Listener dipicu sekali untuk status awal data, dan dipicu kembali setiap kali data berubah. Untuk membaca data di lokasi dan mendeteksi perubahan, digunakan metode

*on()* atau *once()* dari *firebase.database.Reference* untuk mengamati peristiwa.

### 3.6.3.3 Menulis Data Pada Database Melalui Website



Gambar 3. 43 Flowchart Pengiriman Data ke Database Melalui Website

Perancangan ini bertujuan agar mampu memasukan *setpoint* melalui *website*. *Setpoint* yang dimasukan akan tersimpan pada *database*. Untuk dapat menulis data pada *database* juga harus mendapatkan refrensi dari *database*.

Data Firebase diambil dengan menambahkan *listener* asinkron ke *firebase.database.Reference*. Listener dipicu sekali untuk status awal data, dan dipicu kembali setiap kali data berubah. Untuk operasi tulis dasar, menggunakan *set()* untuk menyimpan data ke referensi yang ditentukan, sehingga menggantikan data yang ada di lokasi tersebut.

#### **3.6.3.4 Menampilkan Data Pada Tabel *Website***

Fungsi menampilkan data pada tabel juga merupakan fungsi membaca data pada *database*. Sehingga untuk dapat membaca data dari *database* diperlukan untuk mendapatkan referensi.

-----Halaman ini sengaja dikosongkan-----

## **BAB IV**

### **PENGUKURAN DAN ANALISA**

Untuk mengetahui bahwa hasil dari realisasi sistem yang telah dirancang telah memenuhi kebutuhan perancangan. Pengujian dilakukan dengan melakukan pengukuran dan analisa. Adapun kebutuhan dalam Proyek Akhir ini meliputi pengujian *hardware* dan *software*.

#### **4.1 Hasil Perancangan Hardware**

Hasil perancangan hardware ini meliputi hasil perancangan mekanik dan perancangan elektronik.

##### **4.1.1 Hasil Perancangan Mekanik**

Hasil perancangan mekanik harus sesuai dengan desain yang dibuat baik secara dimensi maupun peletakan komponen yang dibutuhkan sistem. Sehingga setiap subsistem mampu bekerja sesuai dengan perancangan awal.

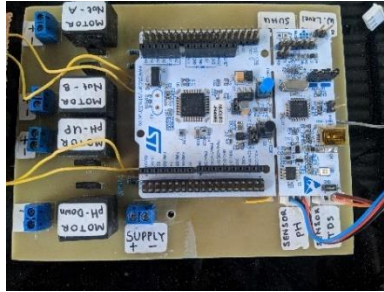


Gambar 4. 1 Hasil Perancangan Mekanik

Gambar 4.1 merupakan hasil realisasi dari desain dan perancangan mekanik yang berfungsi untuk peletakan dari setiap subsistem.

#### 4.1.2 Hasil Perancangan Elektronik

Untuk realisasi perancangan elektronik didahulukan untuk pencetakan desain PCB pada PCB *board*.



Gambar 4. 2 Desain *Layout* PCB

Gambar 4.2 merupakan desain yang dicetak pada PCB. Pencetakan PCB ditujukan untuk peletakan komponen komponen dalam perancangan elektrik sistem.

#### 4.2 Pengujian Subsystem

Untuk setiap subsystem dan komponen hasil perancangan dilakukan pengujian dan pengambambilan data untuk melihat apakah setiap subistsem mampu bekerja sesuai dengan perancangan awal dan dilakukan analisa apakah setiap susbsistem.

##### 4.2.1 Pengujian *Setup* Subsystem Pengendali

Pengujian dilakukan untuk mengetahui apakah pin-pin pada STM32 F446RE NUCLEO bekerja sesuai dengan fungsinya yang dibutuhkan dalam Proyek Akhir.

Salah satu fungsi dari STM32 yang dibutuhkan pada Proyek Akhir ini adalah digital *output*.

Pada pengujian dilakukan pengukuran tegangan pada pin yang akan digunakan pada saat kondisi pin *set* dan *reset*. Pada Proyek Akhir ini kondisi *set* yang dibutuhkan tegangan sebesar 3 ~ 3,3 VDC sedangkan untuk kondisi *reset* dibutuhkan tegangan sebesar 0-1,2 VDC. Program pada STM32 yang diguna untuk melakukan pengujian pin digital dengan kondisi pin *set* yaitu sesuai pada Gambar 4.3. sedangkan untuk program memberikan kondisi *reset* pada pin digital yaitu sesuai dengan Gambar 4.4. Untuk rangkaian pengukuran

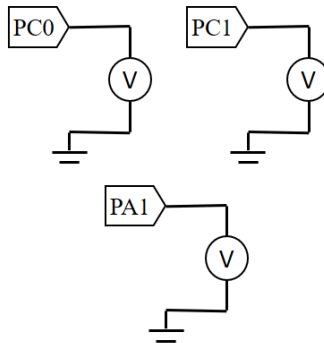
tegangan pada Pin digital STM32 sesuai dengan rangkaian yang ada pada Gambar 4.4.

```
HAL_GPIO_WritePin(GPIOA, Pin_1 , GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOC, Pin_0 , GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOC, Pin_1 , GPIO_PIN_SET);
```

Gambar 4. 3 Program Pin Set Pada STM32

```
HAL_GPIO_WritePin(GPIOA, Pin_1 , GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOC, Pin_0 , GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOC, Pin_1 , GPIO_PIN_RESET);
```

Gambar 4. 4 Program Pin Reset Pada STM32



Gambar 4. 5 Rangkaian Pengukuran Tegangan Pin STM32

Pada Tabel 4.1 merupakan kondisi pengukuran tegangan pin pin yang dibutuhkan pada saat kondisi *set* dan *reset*.

Tabel 4. 1 Data pengukuran tegangan pin STM32

PIN	Tegangan (V)	
	<i>Set</i>	<i>Reset</i>
PC0	3,2	0,2
PC1	3,3	0,2
PA1	3,2	0,2

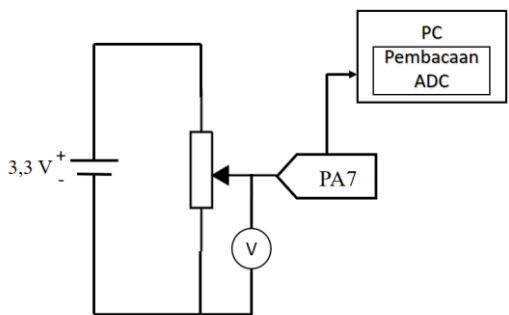
Selain dilakukan pengujian pada digital *output* juga dilakukan pengujian *Analog Digital Converter* (ADC). Pengujian ini dilakukan karena pada Proyek Akhir ini menggunakan sensor analog.

Pengujian dilakukan dengan memberikan tegangan yang bervariasi pada pin ADC STM32 yaitu pin PA7 pemberian tegangan yang bervariasi diperoleh dengan mengubah hambatan pada potensiometer. Dan hasil pembacaan ADC dibandingkan dengan hasil perhitungan. Karena menggunakan 12-bit ADC maka secara perhitungan menggunakan Persamaan 4.1. untuk dapat membaca nilai ADC, di unggah program seperti Gambar 4.6 pada STM32, sedangkan rangkaian pengukuran dilakukan seperti Gambar 4.7

234

HAL\_ADC\_Start\_DMA(&hadc1,(uint32\_t\*)data\_adc,3);

Gambar 4. 6 Program Pembacaan ADC



Gambar 4. 7 Rangkaian Pengujian *Chanel* ADC pada STM32

$$ADC = \frac{V_{in}}{3.3} \times 4095$$

(4.1)

Kemudian data dibandingkan dengan

Pada Tabel 4.2 disajikan data perbandingan data hasil konversi analog ke digital 12-bit STM32 dengan perhitungan setelah dilakukan pengujian.

Tabel 4. 2 Perbandingan Pembacaan ADC STM32 dengan Perhitungan

PIN	Vin (VDC)	ADC		%Error
		STM32	Perhitungan	
PA7	0	4	0	-
PA7	0,28	351	347,4545	1,02%



PIN	Vin (VDC)	ADC		%Error
		STM32	Perhitungan	
PA7	0,53	663	657,6818	0,8%
PA7	0,75	938	930,681	0,78%
PA7	1,1	1372	1365	0,51%
PA7	1,25	1563	1551,136	0,76%
PA7	1,54	1922	1911	0,57%
PA7	1,74	2165	2159,182	0,27%
PA7	1,98	2469	2457	0,48%
PA7	2,25	2807	2792,045	0,53%
PA7	2,48	3099	3077,455	0,7%
PA7	2,76	3437	3424,909	0,35%
PA7	3,12	3891	3871,636	0,5%
PA7	3,27	4068	4057,773	0,25%

Berdasarkan Tabel 4.2 yang merupakan perbandingan antara pembacaan ADC dari STM32 dengan hasil perhitungan. Diperoleh rata-rata %error adalah 0,58%. Dengan %error terbesar adalah 1,02%. Ini menunjukkan bahwa pembacaan ADC masih pada batas toleransi yaitu %error dibawah 5%.

#### 4.2.2 Pengujian Subsistem Sensor

Pengujian yang dilakukan pada sensor TDS yaitu membaca hasil pengukuran sensor menggunakan pin ADC pada pin PA7. Pembacaan tersebut akan di konevrsi berdasarkan spesifikasi yang sesuai dengan sensor TDS *Gravity*. Rumus konversi yang digunakan adalah:

*Output voltage* : 0~2,3 V

*TDS Measurement Range* 0~1000 PPM

Resolusi ADC 12-bit

$$PPM = \left( \frac{3,3}{2,3} \times \frac{1000}{4095} \right) ADC \quad (4.2)$$

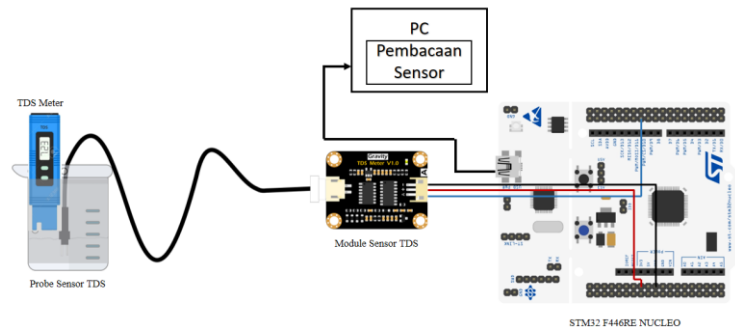
Dari hasil konversi, dilakukan pengujian dengan membandingkan hasil pengukuran sensor TDS setelah dilakukan konversi dengan TDS meter.

Untuk pengujian pengukuran dilakukan dengan mengukur nilai TDS pada air destilasi dengan volume 10,3 L pada wadah yang sama. Dan untuk merubah nilai TDS pada air diberikan sebuah larutan Nutrisi A dan Nutrisi B dengan takaran yang berbeda beda untuk

setiap data. Untuk pembacaan sensor dan koversi ADC ke PPM menggunakan program yang ada pada Gambar 4.8 Pengujian yang dilakukan seperti yang tertera pada Gambar 4.9.

```
281 // Sensor TDS
282 for(int i=0 ; i<250 ; i++)
283 {
284     temp[i] = data_adc[1];
285     ratatds = temp[i]+ratatds;
286     HAL_Delay(2);
287 }
288 nilai[1] = ratatds/250;
289 sensor[1] = ((8e-05)*nilai[1]*nilai[1]) + (0.2731*nilai[1])+27.936;
```

Gambar 4. 8 Program Pembacaan Sensor TDS



Gambar 4. 9 Pengukuran dengan Sensor TDS dan TDS Meter

Berdasarkan pengujian yang dilakukan didapatkan data perbandingan data sensor TDS dengan TDS meter kemudian dicari %error kesalahan menggunakan Persamaan (4.3). dan hasil pengukuran serta perhitungan %error tercantum pada Tabel 4.3

$$\%Error = \frac{|sensor - alat\ ukur|}{alat\ ukur} \times 100\% \tag{4.3}$$

Tabel 4. 3 Perbandingan Pengukuran Sensor TDS dengan TDS Meter

No.	Volume Nutrisi (ml)		Sensor TDS	TDS Meter	%Error
	A	B			
2	0	0	87	89	2,24%

No.	Volume Nutrisi (ml)		Sensor TDS	TDS Meter	%Error
	A	B			
3	1	1	103	107	1,86%
4	3	3	126	130	3,07%
8	4	4	160	159	0,62%
7	4	4	202	199	1,5%
5	5	5	212	214	0,93%
6	6	6	229	227	0,88%
9	3	3	302	293	3,07%
10	2	2	368	369	0,27%
11	1	1	860	862	0,23%
12	2	2	860	862	2,24%

Berdasarkan data pada Tabel 4.4 data perbandingan antara pengukuran sensor TDS dengan alat ukur TDS Meter dapat dicari korelasi antara kedua data tersebut. Korelasi dapat dihitung menggunakan rumus berikut ini:

$$R = \frac{n \sum_{i=1}^n xy - (\sum_{i=1}^n x)(\sum_{i=1}^n y)}{\sqrt{[n \sum_{i=1}^n x^2 - (\sum_{i=1}^n x)^2][n \sum_{i=1}^n y^2 - (\sum_{i=1}^n y)^2]}} \quad (4.4)$$

Setelah dicari korelasi dari kedua data tersebut menggunakan persamaan (4.4), diperoleh korelasi dari sensor TDS dengan TDS meter adalah 0,99. Karena korelasi antara pembacaan sensor TDS dengan TDS meter  $\approx 1$  sehingga pembacaan sensor TDS dapat dikatakan benar. selain itu %error tertinggi berdasarkan Tabel 4.4 adalah 3,07% yang menunjukan bahwa %error pengukuran sensor masih dibawah batas toleransi yaitu 5%.

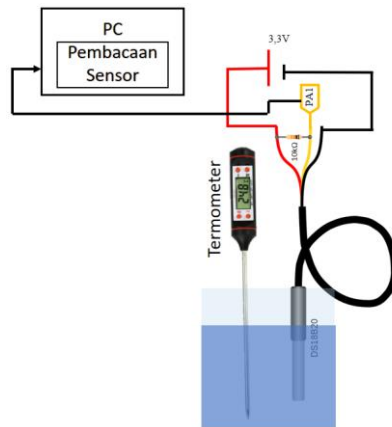
Pengujian pada sensor temperatur DS18B20 dilakukan dengan prosedur yang dilakukan ialah memasukan sensor DS18B20 dan termometer air digital pada wadah yang berisi air. Kemudian mengubah temperatur air pada wadah dengan memanaskan menggunakan *heater*. Sehingga didapatkan data perbandingan pembacaan sensor dengan termometer yang bervariasi. Untuk proses pembacaan sensor dilakukan dengan STM32 F4. Untuk program pembacaan sensor DS18B20 diberikan seperti Gambar 4.10. Sedangkan rangkaian pengujian sensor DS18B20 dilakukan seperti pada Gambar 4.11.

```

304 // Sensor TemperaturDS18B20
305     check = ds18b20_init ();
306         write (0xCC);
307         write (0x44);
308
309         HAL_Delay (800);
310
311         ds18b20_init ();
312         write (0xCC);
313         write (0xBE);
314
315         temp_l = read();
316         temp_h = read();
317         temper = (temp_h<<8)|temp_l;
318         sensor[3] = (float)temper/16;

```

Gambar 4. 10 Program Pembacaan Sensor DS18B20



Gambar 4. 11 Pengukuran Sensor DS18B20 dan Termometer

Kemudia data yang diperoleh dari data perbandingan antara pembacaan sensor dengan termometer dicari persentase kesalahan pembacaan, untuk perhitungan persentase kesalahan perhitungan sesuai dengan Persamaan (4.3).

Tabel 4. 4 Perbandingan Pengukuran Sensor Temperatur DS18B20 dengan Termometer

No	Sensor DS18B20 (°C)	Termometer (°C)	%Error
1	15,1	15,3	1,30%
2	18,7	19,1	2,09%

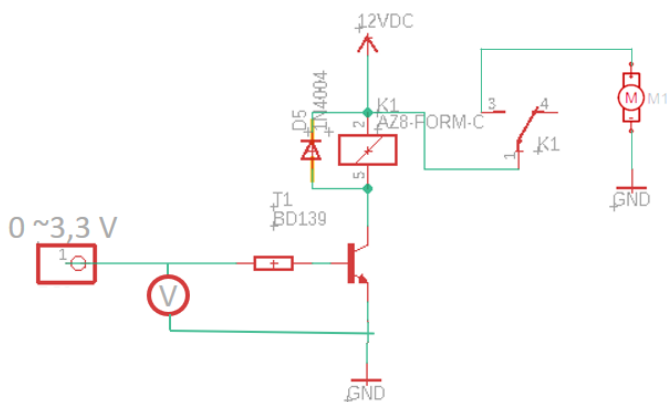
No	Sensor DS18B20 (°C)	Termometer (°C)	%Error
3	23,5	23,7	0,84%
4	25,78	25,9	0,46%
5	27,38	27,9	1,86%
6	33,65	33,5	0,44%
7	41,2	41	0,48%
8	47,68	47,8	0,25%
9	53,9	54,2	0,55%
10	64,8	65,3	0,76%
11	71,65	72,1	0,62%
12	83,51	83,77	0,31%
13	87,1	87	0,11%

Berdasarkan Tabel 4.4 yang merupakan perbandingan hasil pengukuran sensor DS18B20 dengan termometer maka dapat dicari korelasi dari kedua data tersebut menggunakan Persamaan (4.4), diperoleh korelasi dari sensor DS18B20 dengan Termometer adalah 0,99. Karena korelasi antara pembacaan sensor TDS dengan TDS meter  $\approx 1$  sehingga pembacaan sensor DS18B20 dapat dikatakan benar. selain itu %error tertinggi berdasarkan Tabel 4.5 adalah 2,09% yang menunjukkan bahwa %error pengukuran sensor masih dibawah batas toleransi yaitu 5%

#### 4.2.3 Pengujian Subsistem Driver

Pengujian driver dilakukan dengan memberikan tegangan yang berbeda beda pada tegangan  $V_{BB}$  dan melihat kondisi relay, pada tegangan berapakah relay sudah melakukan *switch*. Tujuan dilakukan pengujian untuk menghitung apakah perhitungan yang dilakukan untuk merancang *driver* telah benar. Dan digunakan untuk menganalisa jika tidak terjadi *switching* jika telah diberi kondisi *high*.

Prosedur pengujian yang dilakukan adalah memberikan tegangan  $V_{BB}$  yang berbeda beda mulai 0~3,3 V dan dilihat pada tegangan berapa *relay* aktif. Rangkaian yang digunakan untuk melakukan pengujian *driver* seperti yang ada pada gambar 4.12.



Gambar 4. 12 Pengujian Rangkaian *Driver*

Pada Tabel 4.5 merupakan data membandingkan kondisi relay dengan memberikan tegangan pada  $V_{BB}$  yang berbeda beda.

Tabel 4. 5 Perbandingan  $V_{BB}$  Dengan Kondisi Relay

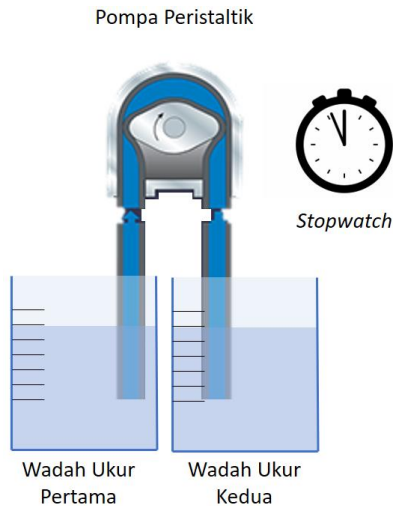
No.	$V_{BB}$ (V)	Kondisi Relay
1	0	<i>Off</i>
2	0,31	<i>Off</i>
3	0,42	<i>Off</i>
4	0,56	<i>Off</i>
5	0,63	<i>Off</i>
6	0,7	<i>Off</i>
7	0,82	<i>Off</i>
8	0,92	<i>Off</i>
9	1,3	<i>Off</i>
10	1,4	<i>Off</i>
11	1,5	<i>Off</i>
12	1,6	<i>Off</i>
13	1,7	<i>Off</i>
14	1,82	<i>Off</i>
15	1,9	<i>Off</i>
16	2,03	<i>On</i>
17	2,11	<i>On</i>
18	2,24	<i>On</i>

Berdasarkan pada Tabel 4.4 menunjukkan bahwa rangkaian *driver* yang telah dirancang dapat digunakan melihat bahwa setelah pengujian pada sub bab 4.2.1 bahwa tegangan keluaran pada pin PC0 dan PC1 bernilai 0,2 V saat kondisi *low* dan bertegangan 3,2 V saat kondisi *high*.

#### 4.2.4 Pengujian Subsistem Aktuator

Pengujian pada pompa peristaltik yang dilakukan adalah mengetahui debit air yang dapat dialirkan pompa peristaltik.

Prosedur pengujian yang dilakukan adalah mencatu pompa peristaltik dengan tegangan 12 VDC saat pompa aktif makan mengaktifkan stopwatch secara bersamaan kemudian pompa akan memindahkan air pada wadah pertama ke wadah kedua. Saat catu daya pompa dilepas maka diperoleh catatan lama waktu pompa menyala serta banyak air yang dapat dipindahkan dari wadah pertama ke wadah kedua. Pengujian dilakukan sesuai pada Gambar 4.13.



Gambar 4. 13 Pengujian Debit Pompa Peristaltik

Dari hasil pengujian dengan mengukur volume air yang dapat dialirkan perwaktunya didapatkan data yang tercantum pada Tabel 4.6.

Tabel 4. 6 Data Pengukuran Volume Air dan Waktu

No.	Waktu (s)	Volume (ml)
1	60	80
2	55	71
3	55	70
4	50	65
5	50	63
6	45	62
7	55	74
8	40	52
9	35	49
10	30	40
11	25	32
12	20	28

Dari hasil pengujian didapatkan data yang tertera pada Tabel 4.6 yang digunakan untuk mendapatkan persamaan debit pompa yaitu pada Persamaan (4.5). Debit pompa dibutuhkan untuk proses pemberian nutrisi.

$$y = 1,303x + 1,0363 \quad (4.5)$$

Di mana :

y = debit air (ml)

x = waktu (*second*)

#### 4.2.5 Pengujian Subsistem *Display*

Pengujian subsistem *display* dilakukan dengan menampilkan karakter sesuai program yang diberikan. Pengujian dilakukan untuk mengetahui apakah setiap bit pada LCD mampu menampilkan karakter dan kesamaan program dengan tampilan LCD.

Pengujian bit pada LCD. Dengan menampilkan “ABCDEFGHJKLMNOPQRST” pada kolom pertama, “UVWXYZabcdefghijklmn” pada kolom kedua, “opqrstuvwxyz12345678” pada kolom ketiga serta “90!@#%&^\*()-=+,:;<>” pada kolom keempat. Program untuk menampilkan karakter pada LCD ditampilkan pada Gambar 4.14 dan Gambar 4.15 merupakan tampilan LCD pengujian pada setiap bit.



```

75 lcd_send_cmd(0x80|0x00);
76 lcd_send_string("ABCDEFGHJKLMNORST");
77
78 lcd_send_cmd(0x80|0x40);
79 lcd_send_string("UVWXYZabcdefghijklmnop");
80
81 lcd_send_cmd(0x80|0x14);
82 lcd_send_string("opqrstuvwxyz12345678");
83
84 lcd_send_cmd(0x80|0x54);
85 lcd_send_string("90!@#%^&*()-=+.,:<>");

```

Gambar 4. 14 Program LCD Pengujian Bit



Gambar 4. 15 Tampilan LCD Pengujian Bit

Pengujian kesamaan tampilan LCD dengan program yang diberikan dengan menampilkan kondisi saat inisial kondisi awal dari masing masing pembacaan sensor. Hasil pemrograman pada LCD ditampilkan pada Gambar 4.7.



Gambar 4. 16 Tampilan LCD Sesuai dengan Inisial Kondisi Awal Sensor

Pengujian kesamaan tampilan LCD dengan program yang diberikan dengan menampilkan hasil pembacaan sensor. Tampilan LCD hasil pemrograman ditampilkan pada Gambar 4.8.



Gambar 4. 17 Tampilan LCD Sesuai dengan Hasil Pengukuran

### 4.3 Pengambilan Data Pemberian Nutrisi

Pengambilan data untuk pemberian nutrisi pada air dilakukan untuk mengetahui kualitas pada setiap air yang berbeda. Serta mengetahui kondisi dan nilai TDS awal pada masing masing jenis air begitupun dengan temperatur air awal.

Pengambilan data juga dilakukan untuk mendapatkan persamaan yang digunakan untu mengethaui perlakuan yang dilakukan untuk pengaturan kadar nutrisi pada air secara otomatis.

Pengambilan data dilakukan pada lima jenis air yang berbeda air yang digunakan adalah air PDAM Surabaya, air destilasi, air Pegunungan Pandaan, air pembuangan *Air Conditioner*(AC), air mineral.

#### 4.3.1 Pemberian Nutrisi Pada Air PDAM Surabaya

Pengambilan data pada air PDAM surabaya dilakukan dengan mengukur nilai TDS dan Temperatur awal. Dan dilakukan pemberian nutrisi A dan nutrisi B dengan takaran yang berbeda beda kemudian mengukur nilai TDS akhir air setelah dilakukan pemberian nutrisi.

Berdasarkan hasil pengukuran untuk air PDAM Surabaya memiliki nilai TDS sebesar 345 PPM Pada Tabel 4.7 ditampilkan data hasil sebelum dan setelah dilakukan pemberian nutrisi A dan nutrisi B yang berbeda beda setiap percobaanya.

Tabel 4. 7 Data TDS Pemberiaan Nutrisi Pada Air PDAM Surabaya

No.	Volume Air(L)	TDS Awal (PPM)	Volume Nutrisi A (mL)	Volume Nutrisi B (mL)	TDS Akhir (PPM)
1	12	345	4	4	441
2	12	345	15	15	739
3	12	363	18	18	821

No.	Volume Air(L)	TDS Awal (PPM)	Volume Nutrisi A (mL)	Volume Nutrisi B (mL)	TDS Akhir (PPM)
4	14,8	363	10	10	473
5	14,8	363	15	15	598
6	14	374	1	1	384
7	14	384	2	2	400
8	14	400	4	4	435
9	14	435	3	3	457
10	14	452	5	5	500

Berdasarkan data pada Tabel 4.7 dipilih 3 data untuk menentukan persamaan yang digunakan sebagai pengaturan nutrisi pada hidroponik secara otomatis, persamaan tersebut yang kemudian akan dimasukkan pada pemrograman STM32.

Dari ketiga data tersebut setiap data diubah menjadi persamaan dengan 3 variabel. Karena perlakuan yang dilakukan untuk pemberian nutrisi A dan nutrisi B harus selalu sama maka volume nutrisi A dan nutrisi B dijadikan satu variabel. Persamaan tersebut adalah

$$12a + 345b + 441c = 8 \quad (4.6)$$

$$12a + 345b + 739c = 30 \quad (4.7)$$

$$14,8a + 363b + 20c = 20. \quad (4.8)$$

Dari persamaan tersebut dicari nilai a,b dan c. Untuk penyelesaiannya menggunakan matriks dengan mencari determinan dari matrik. Berikut adalah perhitungan untuk penyelesaiannya:

$$\begin{bmatrix} a & b & c \\ 12 & 363 & 441 \\ 12 & 345 & 739 \\ 14,8 & 363 & 473 \end{bmatrix} = \begin{bmatrix} 8 \\ 30 \\ 20 \end{bmatrix}$$

$$D = \begin{vmatrix} 12 & 363 & 441 \\ 12 & 345 & 739 \\ 14,8 & 363 & 473 \end{vmatrix} = 223500$$

$$Da = \begin{vmatrix} 8 & 363 & 441 \\ 30 & 345 & 739 \\ 20 & 363 & 473 \end{vmatrix} = 1122564$$

$$Db = \begin{vmatrix} 12 & 8 & 441 \\ 12 & 30 & 739 \\ 14,8 & 20 & 473 \end{vmatrix} = -54954$$

$$Dc = \begin{vmatrix} 12 & 363 & 8 \\ 12 & 345 & 30 \\ 14,8 & 363 & 20 \end{vmatrix} = 223500$$

$$a = \frac{Da}{D} = 5,00268$$

$$b = \frac{Db}{D} = -0,24588$$

$$c = \frac{Dc}{D} = 0,073826$$

Dari perhitungan didapatkan sebuah persamaan yang mana akan digunakan untuk menentukan berapa banyak nutrisi A dan nutrisi B yang digunakan untuk mampu memenuhi *setpoint* yang telah ditentukan. Berikut adalah persamaan yang didapatkan.

$$z = 5,00268w - 0,24588x + 0,073826y \quad (4.9)$$

Di mana :

$z$  = vol. Nutrisi A + Nutrisi B (ml)

$w$  = Vol. Air pada tandon (liter)

$x$  = TDS air (ppm)

$y$  = TDS yang diinginkan (ppm)

#### 4.3.2 Pemberian Nutrisi Pada Air Destilasi

Pengambilan data pada air destilasi dilakukan dengan mengukur nilai TDS dan Temperatur awal. Dan dilakukan pemberian nutrisi A dan nutrisi B dengan takaran yang berbeda beda kemudian mengukur nilai TDS akhir air setelah dilakukan pemberian nutrisi.

Berdasarkan hasil pengukuran untuk air destilasi memiliki nilai TDS sebesar 24 PPM Pada Tabel 4.8 ditampilkan data hasil

sebelum dan setelah dilakukan pemberian nutrisi A dan nutrisi B yang berbeda beda setiap percobaanya.

Tabel 4. 8 Data TDS Pemberiaan Nutrisi Pada Air Destilasi

No.	Volume Air(L)	TDS Awal (PPM)	Volume Nutrisi A (mL)	Volume Nutrisi b (mL)	TDS Akhir (PPM)
1	9,362	168	2	2	185
2	9,362	185	6	6	226
3	9,362	226	4	4	265
4	9,362	265	2	2	289
5	9,362	289	8	8	329
6	9,362	329	3	3	383
7	9,362	383	5	5	418
8	9,362	418	5	5	468
9	9,362	468	3	3	521
10	9,362	521	1	1	534

Berdasarkan data pada Tabel 4.7 dipilih 3 data untuk menentukan persamaan yang digunakan sebagai pengaturan nutrisi pada hidroponik secara otomatis, persamaan tersebut yang kemudian akan dimasukkan pada pemrograman STM32.

Dari ketiga data tersebut setiap data diubah menjadi persamaan dengan 3 variabel. Karena perlakuan yang dilakukan untuk pemberian nutrisi A dan nutrisi B harus selalu sama maka volume nutrisi A dan nutrisi B dijadikan satu variabel. Persamaan tersebut adalah.

$$9,362a + 168b + 185c = 4 \quad (4.10)$$

$$9,362a + 289b + 329c = 16 \quad (4.11)$$

$$9,362a + 418b + 462c = 10 \quad (4.12)$$

Dari persamaan tersebut dicari nilai a,b dan c. Untuk penyelesaiannya menggunakan matriks dengan mencari determinan dari matrik. Berikut adalah perhitungan untuk penyelesaiannya:

$$\begin{matrix} a & b & c \\ \begin{bmatrix} 9,362 & 168 & 185 \\ 9,362 & 289 & 329 \\ 9,362 & 418 & 462 \end{bmatrix} & = & \begin{bmatrix} 4 \\ 16 \\ 10 \end{bmatrix} \end{matrix}$$

$$D = \begin{vmatrix} 9,362 & 168 & 185 \\ 9,362 & 289 & 329 \\ 9,362 & 418 & 462 \end{vmatrix} = -128568,346$$

$$Da = \begin{vmatrix} 4 & 168 & 185 \\ 16 & 289 & 329 \\ 10 & 418 & 462 \end{vmatrix} = -2162$$

$$Db = \begin{vmatrix} 9,362 & 4 & 185 \\ 9,362 & 16 & 329 \\ 9,362 & 10 & 462 \end{vmatrix} = 20502,78$$

$$Dc = \begin{vmatrix} 9,362 & 168 & 4 \\ 9,362 & 289 & 16 \\ 9,362 & 418 & 10 \end{vmatrix} = -21289,188$$

$$a = \frac{Da}{D} = 0,016815959$$

$$b = \frac{Db}{D} = -0,15946989$$

$$c = \frac{Dc}{D} = 0,165586543$$

Dari perhitungan didapatkan sebuah persamaan yang mana akan digunakan untuk menentukan berapa banyak nutrisi A dan nutrisi B yang digunakan untuk mampu memenuhi *setpoint* yang telah ditentukan. Berikut adalah persamaan yang didapatkan

$$z = 0,01681w - 0,1595x + 0,1655y \quad (4.13)$$

Di mana :

$z$  = vol. Nutrisi A + Nutrisi B (ml)

$w$  = Vol. Air pada tandon (liter)

$x$  = TDS air (ppm)

$y$  = TDS yang diinginkan (ppm)

#### 4.3.3 Pemberian Nutrisi Pada Air Pandaan

Pengambilan data pada air pandaan dilakukan dengan mengukur nilai TDS dan Temperatur awal. Dan dilakukan pemberian nutrisi A dan nutrisi B dengan takaran yang berbeda beda kemudian mengukur nilai TDS akhir air setelah dilakukan pemberian nutrisi.

Berdasarkan hasil pengukuran untuk air pandaan memiliki nilai TDS sebesar 168 PPM Pada Tabel 4.9 ditampilkan data hasil sebelum dan setelah dilakukan pemberian nutrisi A dan nutrisi B yang berbeda beda setiap percobaanya.

Tabel 4. 9 Data TDS Pemberiaan Nutrisi Pada Air Pandaan

No.	Volume Air(L)	TDS Awal (PPM)	Volume Nutrisi A (mL)	Volume Nutrisi B (mL)	TDS Akhir (PPM)
1	12	243	1	1	258
2	12	258	2	2	275
3	12	275	3	3	302
4	12	302	4	4	340
5	12	340	5	5	387
6	12	387	6	6	443
7	11	344	1	1	365
8	11	365	2	2	378
9	11	378	3	3	387
10	11	387	4	4	402

Berdasarkan data pada Tabel 4.7 dipilih 3 data untuk menentukan persamaan yang digunakan sebagai pengaturan nutrisi pada hidroponik secara otomatis, persamaan tersebut yang kemudian akan dimasukan pada pemrograman STM32.

Dari ketiga data tersebut setiap data diubah menjadi persamaan dengan 3 variabel. Karena perlakuan yang dilakukan untuk pemberian nutrisi A dan nutrisi B harus selalu sama maka volume nutrisi A dan nutrisi B dijadikan satu variabel. Persamaan tersebut adalah

$$12a + 258b + 275c = 4 \quad (4.14)$$

$$12a + 340b + 387c = 10 \quad (4.15)$$

$$11a + 365b + 378c = 4 \quad (4.16)$$

Dari persamaan tersebut dicari nilai a,b dan c. Untuk penyelesaiannya menggunakan matriks dengan mencari determinan dari matrik. Berikut adalah perhitungan untuk penyelesaiannya:

$$\begin{matrix} a & b & c \\ \begin{bmatrix} 12 & 258 & 275 \\ 12 & 340 & 387 \\ 11 & 365 & 378 \end{bmatrix} & = & \begin{bmatrix} 4 \\ 10 \\ 4 \end{bmatrix} \end{matrix}$$

$$D = \begin{vmatrix} 12 & 258 & 275 \\ 12 & 340 & 387 \\ 11 & 365 & 378 \end{vmatrix} = -128568,346$$

$$Da = \begin{vmatrix} 4 & 258 & 275 \\ 10 & 340 & 387 \\ 4 & 365 & 378 \end{vmatrix} = -2162$$

$$Db = \begin{vmatrix} 12 & 4 & 275 \\ 12 & 10 & 387 \\ 11 & 4 & 378 \end{vmatrix} = 20502,78$$

$$Dc = \begin{vmatrix} 12 & 258 & 4 \\ 12 & 340 & 10 \\ 11 & 365 & 4 \end{vmatrix} = -21289,188$$

$$a = \frac{Da}{D} = 0,016815959$$

$$b = \frac{Db}{D} = -0,15946989$$

$$c = \frac{Dc}{D} = 0,165586543$$

Dari perhitungan didapatkan sebuah persamaan yang mana akan digunakan untuk menentukan berapa banyak nutrisi A dan nutrisi B yang digunakan untuk mampu memenuhi *setpoint* yang telah ditentukan. Berikut adalah persamaan yang didapatkan

$$z = 0,01681w - 0,1595x + 0,1655y \quad (4.17)$$

Di mana:

$$z = \text{vol. Nutrisi A} + \text{Nutrisi B (ml)}$$



$w$  = Vol. Air pada tandon (liter)

$x$  = TDS air (ppm)

$y$  = TDS yang diinginkan (ppm)

#### 4.3.4 Pemberian Nutrisi Pada Air Pendingin Ruangan

Pengambilan data pada air pendingin ruangan dilakukan dengan mengukur nilai TDS dan Temperatur awal. Dan dilakukan pemberian nutrisi A dan nutrisi B dengan takaran yang berbeda beda kemudian mengukur nilai TDS akhir air setelah dilakukan pemberian nutrisi.

Berdasarkan hasil pengukuran untuk air pendingin ruangan memiliki nilai TDS sebesar 24 PPM Pada Tabel 4.10 ditampilkan data hasil sebelum dan setelah dilakukan pemberian nutrisi A dan nutrisi B yang berbeda beda setiap percobaanya.

Tabel 4. 10 Data TDS Pemberiaan Nutrisi Pada Air AC

No.	Volume Air(L)	TDS Awal (PPM)	Volume Nutrisi A (mL)	Volume Nutrisi B (mL)	TDS Akhir (PPM)
1	11,6	188	1	1	200
2	11,6	200	2	2	217
3	11,6	217	3	3	243
4	11,6	243	4	4	279
5	11,6	279	5	5	325
6	11,6	325	6	6	383
7	11,6	383	1	1	391
8	11,6	391	2	2	415
9	11,6	415	3	3	447
10	11,6	447	4	4	483

Berdasarkan data pada Tabel 4.7 dipilih 3 data untuk menentukan persamaan yang digunakan sebagai pengaturan nutrisi pada hidroponik secara otomatis, persamaan tersebut yang kemudian akan dimasukkan pada pemrograman STM32.

Dari ketiga data tersebut setiap data diubah menjadi persamaan dengan 3 variabel. Karena perlakuan yang dilakukan untuk pemberian nutrisi A dan nutrisi B harus selalu sama maka volume nutrisi A dan nutrisi B dijadikan satu variabel. Persamaan tersebut adalah

$$11,6a + 188b + 200c = 2 \quad (4.18)$$

$$11,6a + 243b + 279c = 8 \quad (4.19)$$

$$11,6a + 391b + 415c = 4 \quad (4.20)$$

Dari persamaan tersebut dicari nilai a,b dan c. Untuk penyelesaiannya menggunakan matriks dengan mencari determinan dari matrik. Berikut adalah perhitungan untuk penyelesaiannya:

$$\begin{matrix} a & b & c \\ \begin{bmatrix} 11,6 & 188 & 200 \\ 11,6 & 243 & 279 \\ 11,6 & 391 & 415 \end{bmatrix} & = & \begin{bmatrix} 2 \\ 8 \\ 4 \end{bmatrix} \end{matrix}$$

$$D = \begin{vmatrix} 11,6 & 188 & 200 \\ 11,6 & 243 & 279 \\ 11,6 & 391 & 415 \end{vmatrix} = -48859$$

$$Da = \begin{vmatrix} 2 & 188 & 200 \\ 8 & 243 & 279 \\ 4 & 391 & 415 \end{vmatrix} = 360$$

$$Db = \begin{vmatrix} 11,6 & 2 & 200 \\ 11,6 & 8 & 279 \\ 11,6 & 4 & 415 \end{vmatrix} = 13131,2$$

$$Dc = \begin{vmatrix} 11,6 & 188 & 2 \\ 11,6 & 243 & 8 \\ 11,6 & 391 & 4 \end{vmatrix} = -12852,8$$

$$a = \frac{Da}{D} = -0,00737$$

$$b = \frac{Db}{D} = -0,26876$$

$$c = \frac{Dc}{D} = 0,263058$$

Dari perhitungan didapatkan sebuah persamaan yang mana akan digunakan untuk menentukan berapa banyak nutrisi A dan nutrisi B yang digunakan untuk mampu memenuhi *setpoint* yang telah ditentukan. Berikut adalah persamaan yang didapatkan

$$z = -0,00737w - 0,26876x + 0,263058y \quad (4.21)$$

Di mana :

$z$  = vol. Nutrisi A + Nutrisi B (ml)

$w$  = Vol. Air pada tandon (liter)

$x$  = TDS air (ppm)

$y$  = TDS yang diinginkan (ppm)

#### 4.3.5 Pemberian Nutrisi Pada Air Mineral

Pengambilan data pada air mineral dilakukan dengan mengukur nilai TDS dan Temperatur awal. Dan dilakukan pemberian nutrisi A dan nutrisi B dengan takaran yang berbeda beda kemudian mengukur nilai TDS akhir air setelah dilakukan pemberian nutrisi. Berdasarkan hasil pengukuran untuk air mineral memiliki nilai TDS sebesar 208 PPM Pada Tabel 4.11 ditampilkan data hasil sebelum dan setelah dilakukan pemberian nutrisi A dan nutrisi B yang berbeda beda setiap percobaanya.

Tabel 4. 11 Data TDS Pemberiaan Nutrisi Pada Air Mineral

No.	Volume Air(L)	TDS Awal (PPM)	Volume Nutrisi A (mL)	Volume Nutrisi B (mL)	TDS Akhir (PPM)
1	9,22	208	1	1	215
2	9,22	215	2	2	240
3	9,22	240	3	3	275
4	7,8	275	4	4	327
5	7,8	327	5	5	391
6	7,8	391	6	6	477
7	7,8	477	1	1	493
8	7,8	493	2	2	515
9	7,8	515	3	3	557
10	7,8	557	4	4	602

Berdasarkan data pada Tabel 4.11 dipilih 3 data untuk menentukan persamaan yang digunakan sebagai pengaturan nutrisi pada hidroponik secara otomatis, persamaan tersebut yang kemudian akan dimasukan pada pemrograman STM32.

Dari ketiga data tersebut setiap data diubah menjadi persamaan dengan 3 variabel. Karena perlakuan yang dilakukan untuk

pemberian nutrisi A dan nutrisi B harus selalu sama maka volume nutrisi A dan nutrisi B dijadikan satu variabel. Persamaan tersebut adalah

$$9,22a + 215b + 240c = 4 \quad (4.22)$$

$$7,8a + 327b + 391c = 10 \quad (4.23)$$

$$7,8a + 493b + 515c = 4 \quad (4.24)$$

Dari persamaan tersebut dicari nilai a,b dan c. Untuk penyelesaiannya menggunakan matriks dengan mencari determinan dari matrik. Berikut adalah perhitungan untuk penyelesaiannya:

$$\begin{matrix} a & b & c \\ \begin{bmatrix} 9,22 & 215 & 240 \\ 7,8 & 327 & 391 \\ 7,8 & 493 & 515 \end{bmatrix} & = & \begin{bmatrix} 4 \\ 10 \\ 4 \end{bmatrix} \end{matrix}$$

$$D = \begin{vmatrix} 9,22 & 215 & 240 \\ 7,8 & 327 & 391 \\ 7,8 & 493 & 515 \end{vmatrix} = -121777$$

$$Da = \begin{vmatrix} 4 & 215 & 240 \\ 16 & 327 & 391 \\ 4 & 493 & 515 \end{vmatrix} = 858$$

$$Db = \begin{vmatrix} 9,22 & 4 & 240 \\ 7,8 & 10 & 391 \\ 7,8 & 4 & 515 \end{vmatrix} = 17962,12$$

$$Dc = \begin{vmatrix} 9,22 & 215 & 4 \\ 7,8 & 327 & 10 \\ 7,8 & 493 & 4 \end{vmatrix} = -18153,6$$

$$a = \frac{Da}{D} = -0,00705$$

$$b = \frac{Db}{D} = -0,1475$$

$$c = \frac{Dc}{D} = 0,149072$$

Dari perhitungan didapatkan sebuah persamaan yang mana akan digunakan untuk menentukan berapa banyak nutrisi A dan nutrisi B yang digunakan untuk mampu memenuhi *setpoint* yang telah ditentukan. Berikut adalah persamaan yang didapatkan

$$z = -0,00705w - 0,1475x + 0,149072y \quad (4.25)$$

Di mana :

$z$  = vol. Nutrisi A + Nutrisi B (ml)

$w$  = Vol. Air pada tandon (liter)

$x$  = TDS air (ppm)

$y$  = TDS yang diinginkan (ppm)

#### 4.4 Pengujian Ketepatan Pengendalian Nutrisi

Pengujian dilakukan ketepatan pengaturan merupakan pengujian untuk memastikan bahwa sistem yang dirancang mampu mengatur kadar nutrisi dengan tepat. Pengujian ini juga dilakukan untuk mengetahui persamaan yang tepat untuk melakukan pengaturan.

##### 4.4.1 Pengujian Pengaturan Nutrisi pada Air Pandaan

Prosedur pengujian dilakukan dengan memberikan *setpoint* pada setiap pengukuran dan pengujian dilakukan menggunakan ke empat persamaan yang diperoleh pada pengujian yang dilakukan pada sub bab 4.2 yang tepat untuk melakukan pengaturan pada air pandaan.

Tabel 4. 12 Perbandingan *Setpoint* dengan TDS Akhir Pada Air Pandaan

Persamaan	TDS Awal	<i>Setpoint</i>	TDS Akhir	%Error
(4.7)	163	180	240	33,33%
(4.7)	295	310	319	2,90%
(4.7)	319	321	328	2,18%
(4.7)	336	367	344	6,26%
(4.7)	344	367	374	1,9%
(4.11)	391	410	412	0,48%
(4.11)	430	450	451	0,22%
(4.11)	465	479	481	0,41%
(4.11)	498	515	519	0,77%
(4.11)	519	531	535	0,75%
(4.19)	535	556	565	1,61%
(4.19)	570	581	588	1,20%

Persamaan	TDS Awal	<i>Setpoint</i>	TDS Akhir	%Error
(4.19)	588	603	593	1,65%
(4.19)	593	603	616	2,15%
(4.19)	616	634	633	0,15%

Dari hasil pengaturan menggunakan tiga macam persamaan, maka nilai TDS setelah pengaturan dibandingkan dengan nilai *setpoint*. Dari hasil perbandingan tersebut ditemukan *error* dari masing masing pengaturan.

Berdasarkan Tabel 4. Error dari persamaan yang digunakan untuk mengatur kadar nutrisi adalah sebagai berikut:

Untuk pengendalian nutrisi menggunakan persamaan (4.7) rata-rata error yang diperoleh adalah 0,78%. Dan untuk pengendalian nutrisi menggunakan persamaan (4.11) rata-rata %*error* yang diperoleh adalah 0,71%. Sedangkan pengendalian (4.19) rata-rata %*error* yang diperoleh adalah 1,35%.

Sehingga untuk pengendalian nutrisi pada jenis air pandaan untuk mendapatkan hasil pengendalian yang sesuai dengan *setpoint* yang diberikan adalah menggunakan Persamaan (4.13).

#### 4.4.2 Pengujian Pengaturan Nutrisi pada Air PDAM Surabaya

Prosedur pengujian dilakukan dengan memberikan *setpoint* pada setiap pengukuran dan pengujian dilakukan menggunakan ke empat persamaan yang diperoleh pada pengujian yang dilakukan pada sub bab 4.2 yang tepat untuk melakukan pengaturan pada air pandaan.

Tabel 4. 13 Perbandingan *Setpoint* dengan TDS Akhir Pada Air

Persamaan	TDS Awal	<i>Setpoint</i>	TDS Akhir	%Error
(4.9)	332	345	357	3,47%
(4.9)	358	376	391	3,98%
(4.9)	395	408	417	2,20%
(4.9)	419	435	452	3,90%
(4.9)	453	471	479	1,69%
(4.13)	481	503	506	0,59%
(4.13)	506	517	521	0,77%
(4.13)	521	537	541	0,74%
(4.13)	539	553	555	0,36%
(4.13)	555	571	577	1,05%
(4.21)	577	583	580	0,51%
(4.21)	580	583	589	1,02%

Persamaan	TDS Awal	Setpoint	TDS Akhir	%Error
(4.21)	588	612	615	0,49%
(4.21)	615	629	635	0,95%
(4.21)	635	652	656	0,61%

Dari hasil pengaturan menggunakan tiga macam persamaan, maka nilai TDS setelah pengaturan dibandingkan dengan nilai *setpoint*. Dari hasil perbandingan tersebut ditemukan *error* dari masing masing pengaturan.

Berdasarkan Tabel 4. Error dari persamaan yang digunakan untuk mengatur kadar nutrisi adalah sebagai berikut:

Untuk pengendalian nutrisi menggunakan Persamaan (4.9) rata-rata error yang diperoleh adalah 3,05%. Dan untuk pengendalian nutrisi menggunakan Persamaan (4.13) rata-rata %*error* yang diperoleh adalah 0,70%. Sedangkan pengendalian dengan Persamaan (4.21) rata-rata %*error* yang diperoleh adalah 0,72%.

Sehingga untuk pengendalian nutrisi pada jenis air pandaan untuk mendapatkan hasil pengendalian yang sesuai dengan *setpoint* yang diberikan adalah menggunakan Persamaan (4.13).

#### 4.4.3 Pengujian Pengaturan Nutrisi pada Air Destilasi

Prosedur pengujian dilakukan dengan memberikan *setpoint* pada setiap pengukuran dan pengujian dilakukan menggunakan ke empat persamaan yang diperoleh pada pengujian yang dilakukan pada sub bab 4.2 yang tepat untuk melakukan pengaturan pada air pandaan.

Tabel 4. 14 Perbandingan *Setpoint* dengan TDS Akhir Pada Air Destilasi

Persamaan	TDS Awal	Setpoint	TDS Akhir	%Error
(4.9)	54	67	75	11,94%
(4.9)	75	95	104	9,47%
(4.9)	104	137	161	17,51%
(4.9)	161	174	179	2,87%
(4.9)	179	187	194	3,74%
(4.13)	194	124	127	2,41%
(4.13)	127	139	145	4,31%
(4.13)	465	479	479	0%
(4.13)	479	488	486	0,40%
(4.13)	486	488	494	1,22%
(4.21)	535	555	565	1,80%

Persamaan	TDS Awal	<i>Setpoint</i>	TDS Akhir	%Error
(4.21)	575	581	587	1,03%
(4.21)	587	607	611	0,65%
(4.21)	611	625	630	0,8%
(4.21)	630	637	642	0,78%

Dari hasil pengaturan menggunakan tiga macam persamaan, maka nilai TDS setelah pengaturan dibandingkan dengan nilai *setpoint*. Dari hasil perbandingan tersebut ditemukan *error* dari masing masing pengaturan.

Berdasarkan Tabel 4.14 Error dari persamaan yang digunakan untuk mengatur kadar nutrisi adalah sebagai berikut:

Untuk pengendalian nutrisi menggunakan Persamaan (4.9) rata-rata error yang diperoleh adalah 9,1%. Dan untuk pengendalian nutrisi menggunakan Persamaan (4.13) rata-rata %*error* yang diperoleh adalah 1,67%. Sedangkan pengendalian (4.21) rata-rata %*error* yang diperoleh adalah 1,01%.

Sehingga untuk pengendalian nutrisi pada jenis air pandaan untuk mendapatkan hasil pengendalian yang sesuai dengan *setpoint* yang diberikan adalah Persamaan (4.21)

#### 4.4.4 Pengujian Pengaturan Nutrisi pada Air Mineral

Prosedur pengujian dilakukan dengan memberikan *setpoint* pada setiap pengukuran dan pengujian dilakukan menggunakan ke empat persamaan yang diperoleh pada pengujian yang dilakukan pada sub bab 4.2 yang tepat untuk melakukan pengaturan pada air pandaan.

Tabel 4. 15 Perbandingan *Setpoint* dengan TDS Akhir Pada Air Mineral

Persamaan	TDS Awal	<i>Setpoint</i>	TDS Akhir	%Error
(4.9)	236	245	249	33,33%
(4.9)	249	257	263	2,90%
(4.9)	263	275	281	2,18%
(4.9)	281	295	301	6,26%
(4.9)	301	312	316	1,9%
(4.13)	316	323	325	0,48%
(4.13)	325	331	333	0,22%
(4.13)	333	345	348	0,41%
(4.13)	348	357	357	0,77%
(4.13)	357	365	368	0,75%



Persamaan	TDS Awal	Setpoint	TDS Akhir	%Error
(4.21)	368	377	378	1,61%
(4.21)	378	388	392	1,20%
(4.21)	392	407	411	1,65%
(4.21)	411	425	429	2,15%
(4.21)	429	445	449	0,15%

Dari hasil pengaturan menggunakan tiga macam persamaan, maka nilai TDS setelah pengaturan dibandingkan dengan nilai *setpoint*. Dari hasil perbandingan tersebut ditemukan *error* dari masing masing pengaturan.

Berdasarkan Tabel 4.15 Error dari persamaan yang digunakan untuk mengatur kadar nutrisi adalah sebagai berikut:

Untuk pengendalian nutrisi menggunakan persamaan (4.7) rata-rata error yang diperoleh adalah 2,3%. Dan untuk pengendalian nutrisi menggunakan persamaan (4.11) rata-rata %*error* yang diperoleh adalah 0,75%. Sedangkan pengendalian (4.19) rata-rata %*error* yang diperoleh adalah 0,84%.

Sehingga untuk pengendalian nutrisi pada jenis air pandaan untuk mendapatkan hasil pengendalian yang sesuai dengan *setpoint* yang diberikan adalah menggunakan Persamaan (4.13).

## 4.5 Pengujian Website Sebagai Sistem Informasi

*Website* sebagai Sistem Informasi harus mampu menyajikan informasi yang menyatakan kondisi fisik yang ada pada kebun hidroponik terutama pada kondisi air. Mengingat air merupakan media utama dalam berpolacocok tanam hidroponik.

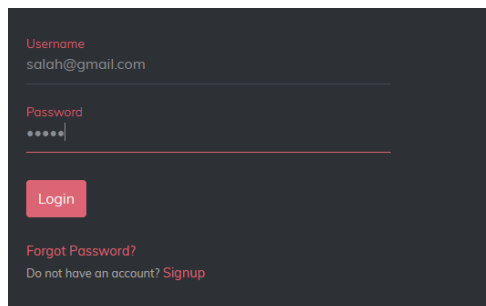
### 4.5.1 Pengujian Ketepatan Autentifikasi Pada Login Page

Pada halaman *login* dilakukan pengujian apakah jika melakukan *login* dengan memasukan data *user* yang terautentifikasi dan data yang tidak terautentifikasi yang telah didaftarkan aksi yang dilakukan oleh *website* sesuai dengan program yang dibuat.

Untuk *username* yang didaftarkan agar dapat terautentifikasi dengan *website* adalah “gilang@gmail.com” sedangkan *password* yang terdaftar adalah “Salah”

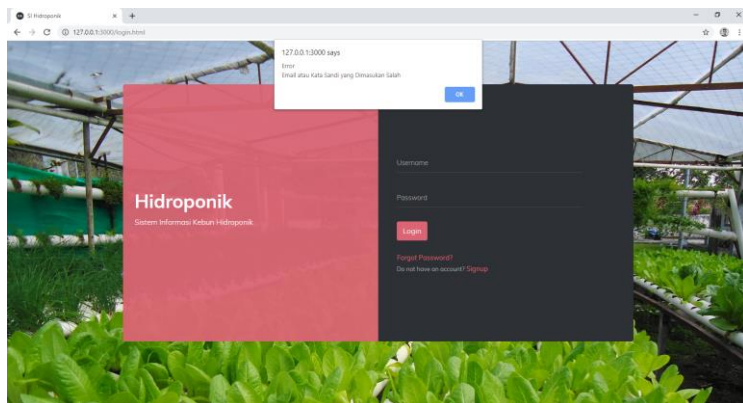
Untuk percobaan yang pertama dilkukan dengan mengisi data yang salah pada *form login*. *Username* yang dimasukan pada *form login* adalah “salah@gmail.com” sedangkan untuk *password*

yang dimasukan adalah “salah”. Gambar 4.9 Menampilkan *form login* dengan pengisian *username* dan *password* yang salah.

A screenshot of a login form on a dark background. The 'Username' field contains 'salah@gmail.com' and the 'Password' field contains six dots. A red 'Login' button is below the fields. Below the button, there is a link for 'Forgot Password?' and a link for 'Do not have an account? Signup'.

Gambar 4. 18Memasukan Data Yang Salah Pada *Form Login*

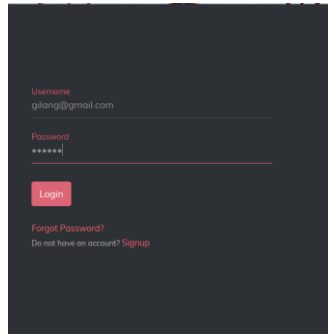
Setelah memasukan data *user* yang salah aksi yang dilakukan adalah *user* tetap berada di halaman *login* dan terdapat sebuah *popup* notifikasi bahwa “Email atau kata sandi yang dimasukan salah”. Tampilan dan aksi yang dilakukan *website* jika data *user* yang dimasukan tidak terautentifikasi ditampilkan pada Gambar 4.10.



Gambar 4. 19 Error Karena Data *Login* Salah

Percobaan kedua yang dilakukan adalah dengan memasukan data *user* yang telah terautentifikasi dengan *website*. *Username* yang dimasukan adalah “gilang@gmail.com” dengan *password* . Gambar

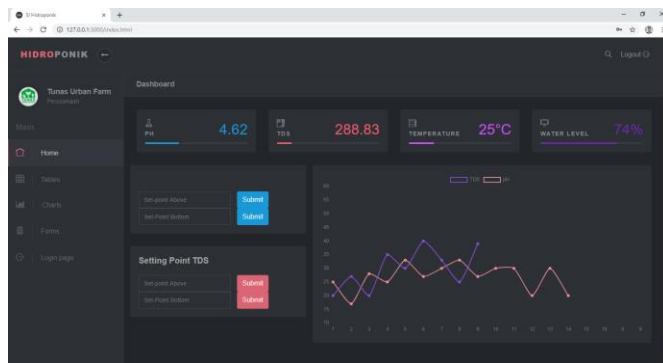
4.7 menampilkan *form login* dengan pengisian *username* dan *password* yang benar.



A screenshot of a login form on a dark background. The 'Username' field contains 'gilang@gmail.com' and the 'Password' field contains '\*\*\*\*\*'. A red 'Login' button is visible below the fields. Below the button, there is a link for 'Forgot Password?' and a link for 'Do not have an account? Signup'.

Gambar 4. 20 Memasukan Data Yang Salah Pada *Form Login*

Setelah terjadi memasukan *username* dan *password* yang dikenali atau yang terautentifikasi oleh *website* aksi yang dilakukan *website* kepada *user* adalah mengalihkan *user* ke halaman utama. Aksi yang dilakukan *website* setelah data yang masukan sesuai dengan data yang terautentifikasi.



Gambar 4. 21 Halaman Utama *Website*

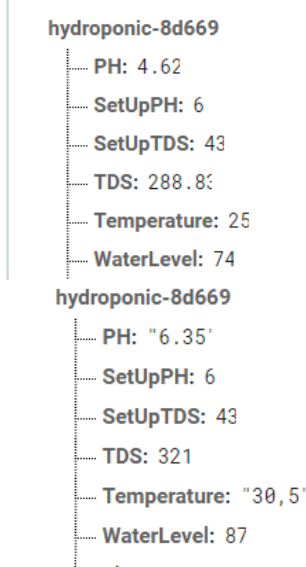
Berdasarkan pengujian yang telah dilakukan aksi yang dilakukan jika data *user* yang masukan salah dan jika data yang dimasukan benar telah sesuai dengan perancangan yang dilakukan untuk *login page*. Yaitu jika akun yang dimasukan tidak

terautentifikasi maka *user* tetap berada pada halaman *login* dan terdapat peringatan bahwa “Email atau Kata Sandi yang Dimasukan Salah”. Sedangkan jika data yang dimasukan oleh *user* telah terautentifikasi, maka *user* akan dialihkan ke halaman muka (*index.html/*).

**4.5.2 Menampilkan Data Sensor Pada Website**

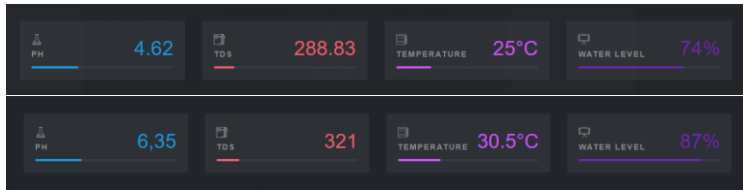
Salah satu fungsi *website* sebagai sistem informasi yaitu mampu menampilkan kondisi pengukuran sensor sesuai dengan data pengukuran sesungguhnya yang ada pada *database*. Data pengukuran akan ditampilkan pada *dashboard* halaman muka. Pengujian dilakukan dengan membandingkan data yang ada pada *database* dengan tampilan pada *website*.

Data hasil pengukuran sensor yang terdapat pada *database* firebase ditunjukkan pada Gambar 4.16.



Gambar 4. 22 Data Pada Database

Sedangkan data yang ditampilkan pada *dashboard* pada *website* berdasarkan data yang terdapat pada data base ditampilkan pada Gambar 4.17.



Gambar 4. 23 Tampilan *Dashboard Website*

Berdasarkan hasil pengujian tersebut menunjukkan bahwa data yang terdapat pada *database* telah dapat ditampilkan dengan benar pada *dashboard website*.

#### 4.5.3 Penulisan Data Pada *Database* Melalui *Website*

Selain mampu menampilkan data hasil pengukuran, website juga dirancang mampu mengatur *setpoint*. Sehingga *website* dirancang mampu menuliskan data pada *database*. Yang kemudian data tersebut akan dibaca oleh NodeMCU. Parameter keberhasilan dari perancangan ini adalah. Data yang di masukan *user* pada *website* harus seusai dengan data yang terdapat pada *database*.

Percobaan dilakukan dengan memasukkan nilai *setpoint* pada *website* kemudian data dibandingkan dengan data yang ada pada *database*. Pada Gambar 4.24 Menampilkan form untuk memasukkan *setpoint* ph. Dan untuk pengujian pertama *setpoint* bawah untuk ph diberikan 5,5 dan untuk *setpoint* bawah ph atas 6,5. Gambar 4.24 Merupakan Form memasukkan *setpoint* untuk batas atas dan bawah pH.

**Setting Point pH**

5.5	Submit
6.5	Submit

Gambar 4. 24 *Form Setup pH*

Untuk pengujian kedua *setpoint* atas untuk TDS diberikan 350 dan untuk *setpoint* bawah TDS diberikan 200. Gambar 4. Merupakan Form memasukkan *setpoint* untuk batas atas dan bawah TDS.

Gambar 4. 25 *Form Setup TDS.*

Sedangkan data yang tersimpan pada *database* ditampilkan pada Gambar. Gambar 4.13

hydroponic-8d669

PH:	"6,35"	×
SetDownPH:	5.5	
SetDownTDS:	200	
SetUpPH:	6.5	
SetUpTDS:	350	
TDS:	321	
Temperature:	"30.5"	
WaterLevel:	87	

Gambar 4. 26 Data Pada *Database* Setelah Dilakukan *Setup* TDS dan pH

#### 4.6 Pengujian Keseluruhan Sistem

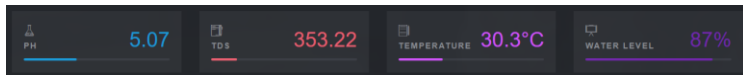
Pengujian keseluruhan dari sistem yang dibuat dilakukan dengan mencoba sistem dari mulai *login* pada *website*, menampilkan data pembacaan sensor serta memasukkan *setpoint* melalui *website* serta ketepatan pengendalian

#### 4.6.1 Pengujian Login User

User melakukan login pada halaman *login*. Halaman login muncul saat membuka alamat *website*. Login dilakukan dengan memasukkan *username* dan *password* yang telah terautentifikasi pada *website*. Pengujian *login* telah dijelaskan pada sub bab

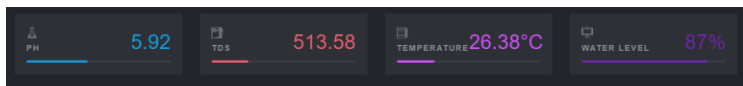
#### 4.6.2 Pengendalian Melalui Website

Pengendalian pH maupun nutrisi dilakukan dengan menentukan *setpoint* yang perlu dicapai dari sistem. Sehingga nutrisi pada hidroponik dapat terpenuhi. Pemberian *setpoint* dilakukan melalui *website*. Kondisi air sebelum dikendalikan adalah seperti yang ditampilkan pada Gambar 4.27.



Gambar 4. 27 Tampilan *Website* Sebelum Pengendalian

Kemudian dilakukan pengendalian dengan memasukkan nilai *setpoint* pada *website* untuk pH nilai yang dimasukan adalah pada batas bawah pH adalah 5,5 dan batas atas untuk pH adalah 6,5. Sedangkan untuk TDS, batas bawah TDS diberikan 500 dan batas atas TDS diberikan 600. Dan hasil dari pengendalian setelah dimasukan *setpoint* untuk masing masing kondisi ditampilkan pada Gambar



Gambar 4. 28 Tampilan *Website* Setelah Pengendalian

#### 4.6.3 Pengujian Sistem Pada Hidroponik

Setelah dilakukan pengujian terhadap keseluruhan subsistem, dengan pembacaan antara sensor dan alat ukur memiliki korelasi 0,99 dan hasil pengendalian yang diperoleh memiliki %*error* rata-rata 0,97% dengan Persamaan (4.13). Dan *website* mampu menampilkan data hasil pengukuran sebagai media pemantauan dan pengendalian air pada kebun hidroponik.

Prosedur pengujian yang dilakukan ialah melakukan pengukuran kondisi air perharinya menggunakan alat hasil

perancangan. Dan hasil pengendalian nutrisi jika kadar nutrisi pada air tidak memenuhi *setpoint*. *Setpoint* yang diinginkan adalah 550-700ppm. Pengujian dilakukan selama 10 hari dengan pemantauan pengukuran dilakukan tiga kali sehari.

Tabel 4. 16 Pengujian Alat Pada Hidroponik

Tanggal	jam	TDS Air (PPM)	Temperatur Air (°C)
12-06-2019	08.00	557,31	26,3
12-06-2019	14.00	580,31	28,43
12-06-2019	19.00	585,77	27,81
13-06-2019	08.00	593,6	24,34
13-06-2019	14.00	572,73	26,74
13-06-2019	19.00	581,07	25,49
14-06-2019	08.00	583,44	25,88
14-06-2019	14.00	598,63	27,3
14-06-2019	19.00	635,88	25,96
15-06-2019	08.00	678,8	26,7
15-06-2019	14.00	556,54	29,91
15-06-2019	19.00	559,79	28,06
16-06-2019	08.00	577,6	26,88
16-06-2019	14.00	573,23	26,66
16-06-2019	19.00	575,05	25,84
17-06-2019	08.00	613,7	25,76
17-06-2019	14.00	618,17	27,22
17-06-2019	19.00	642,3	27,12
18-06-2019	08.00	673,13	26,67
18-06-2019	14.00	562,89	29,22
18-06-2019	19.00	582,97	27,32
19-06-2019	08.00	587,02	26,09
19-06-2019	14.00	573, 42	27,89
19-06-2019	19.00	598,61	26,06
20-06-2019	08.00	591,89	25,73
20-06-2019	14.00	622,32	27,02
20-06-2019	19.00	635,8	26,78
21-06-2019	08.00	677	26,2
21-06-2019	14.00	681,8	29,97
21-06-2019	19.00	687,92	27,65

Tabel 4.16 Merupakan hasil pengujian alat pada kebun hidroponik. Dari hasil pengujian tersebut menunjukan bahwa alat mampu mempertahankan kadar nutrisi pada rentang angka 550-700.



Dan dari pengukuran temperatur juga telah menunjukkan bahwa temperatur air tidak mengalami perubahan yang sehingga menyebabkan temperatur air tidak sesuai dengan *setpoint*.

Dari pengujian tersebut juga didapatkan bahwa penggantian air yang cocok pada hidroponik adalah setiap tiga hari sekali. Terbukti pada data yang terlampir pada Tabel 4.16 bahwa pada hari ke-3 nilai TDS air telah bernilai 635,88ppm, pada hari ke-6 menunjukkan 642,3ppm dan hari ke-9 nilai TDS menunjukkan 635ppm.

-----Halaman ini sengaja dikosongkan-----

## **BAB V**

### **PENUTUP**

Bab ini berisi tentang kesimpulan-kesimpulan yang didapatkan selama proses pembuatan Proyek Akhir ini beserta saran-saran untuk perbaikan dan pengembangannya.

#### **5.1 Kesimpulan**

Berdasarkan pengujian dan pengukuran sistem yang telah dirancang dapat ditarik beberapa kesimpulan diantaranya yaitu sebagai berikut:

1. Hasil pengukuran sensor TDS dan sensor temperatur DS18B20 menunjukkan bahwa sensor telah terkalibrasi dengan benar, karena korelasi yang diperoleh antara sensor dengan TDS meter dan termometer adalah 0,99.
2. Kebutuhan nutrisi pada kebun hidroponik *nutrient film technique* (NFT) dapat terkendali dengan ketepatan melalui pemberian nutrisi A dan nutrisi B, persentase kesalahan pengendalian terbesar yaitu 2,3%.
3. *Website* sebagai sistem informasi mampu memberika informasi mengenai kondisi pH, TDS, temperatur serta *level* air pada kebun hidroponik serta dapat melakukan pengendali untuk merubah nilai pH dan TDS air.

#### **5.2 Saran**

Saran –saran yang diberika berdasarkan hasil pengukuran dan pengujian yang telah dilakukan, agar mampu meningkatkan penelitian selanjutnya adalah sebagai berikut:

1. Menggunakan sensor TDS dengan range pengukuran lebih dari 1000ppm.
2. Menggunakan sensor yang mampu mengukur hanya kadar nutrisi yang ada pada air.
3. Mampu mengendalikan temperatur jika kondisi temperatur tidak sesuai dengan kebutuhan tanaman.
4. Memberi informasi kebutuhan nutrisi maupun pH berdasarkan jenis tanaman.

5. Memberikan peringatan jika temperatur air tidak sesuai dengan kebutuhan tanaman.

## DAFTAR PUSTAKA

- [1] Yos Sutioso, *Hidroponik ala Yos*, Penebar Swadaya, Jakarta, 2004.
- [2] ....., *Datasheet STM32F446RE, STM32F103xE*: STMicroelectronics, 2009.
- [3] Maylita Martani dan Endarko, “Perancangan dan Pembuatan Sesnsor TDS Pada Proses Pengendapan  $CaCO_3$  Dalam Air Dengan Metode Pelucutan Elektron dan Medan Magnet”, *Berkala Fisika*, Institut Teknologi Sepuluh Nopember, Vol 17 No. 3, Hal. 99-108, 2014
- [4] Nurazizah, E.,Ramdhani dan M.,Rizal, A, “Rancang Bangun Termometer Digital Berbasis Sensor DS18B20 Untuk Penyandang Tunanetra”, *E-Proceeding of Engineering*, Vol.4 No.3, Hal 201-2010, 2017.
- [5] Navi, Mahmood, *Rangkaian Listrik*, Penerbit Erlangga, Jakarta, 2004.
- [6] Blum, Jeremy, *Exploring Arduino: Tools and Technique for Engineering Wizardry*, Wiley,Kanada, 2013.
- [7] Bayu Irianto Hadi P, “Rancang Bangun DC Buck Boost Converter Berbasis Arduino Pada Mini Plant Solar Tracking System”, *Tugas Akhir*, Program D3 Metrologi dan Instrumentasi-FTI ITS, Surabaya, 2016.
- [8] Kale, Jacob, *NodeMCU a Complete Guide*, Emereo Pty Limited, Denamrk, 2018.
- [9] Hasim Ashari, “STM32 ARM Coretex-M Sebagai Pembelajaran Mikrokontroler”, *Tugas Akhir Skripsi*, Prohram Studi Pendidikan Teknik Elektronika-FT UNY Yogyakarta, 2018.

-----Halaman ini sengaja dikosongkan-----

# LAMPIRAN A

## A.Datasheet Sensor TDS



### Gravity: Analog TDS Sensor / Meter For Arduino

SKU: SEN0244

TDS (Total Dissolved Solids) indicates that how many milligrams of soluble solids dissolved in one liter of water. In general, the higher the TDS value, the more soluble solids dissolved in water, and the less clean the water is. Therefore, the TDS value can be used as one of the references for reflecting the cleanliness of water.

TDS pen is a widely used equipment to measure TDS value. The price is affordable, and it is easy to use, but it is not able to transmit data to the control system for online monitoring to do some water quality analysis. The professional instrument has high accuracy and can send data to the control system, but the price is expensive for the ordinary people. To this end, we have launched an analog TDS sensor kit which is compatible with Arduino, plug and play, easy to use. Matching with Arduino controller, you can build a TDS detector easily to measure the TDS value of liquid.

This product supports 3.3 ~ 5.5V wide voltage input, and 0 ~ 2.3V analog voltage output, which makes it compatible with 5V or 3.3V control system or board. The excitation source is AC signal, which can effectively prevent the probe from polarization and prolong the life of the probe, meanwhile, increase the stability of the output signal. The TDS probe is waterproof, it can be immersed in water for long time measurement.

---

This product can be used in water quality application, such as domestic water, hydroponics. With this product, you can easily DIY a TDS detector to reflect the cleanliness of water to protect your health.



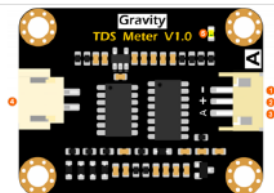
**Attention:**

- 1.The probe can not be used in water above 55 degrees centigrade.
- 2.The probe can not be left too close to the edge of the container, otherwise it will affect the reading.
- 3.The head and the cable of the probe are waterproof, but the connector and the signal transmitter board are not waterproof. Please be careful.

## Specification

- **Signal Transmitter Board**  
Input Voltage: 3.3 ~ 5.5V  
Output Voltage: 0 ~ 2.3V  
Working Current: 3 ~ 8mA  
TDS Measurement Range: 0 ~ 1000ppm  
TDS Measurement Accuracy:  $\pm 10\%$  F.S. (25 °C)  
Module Size: 42 \* 32mm  
Module Interface: PH2.0-3P  
Electrode Interface: XH2.54-2P
- **TDS probe**  
Number of Needle: 2  
Total Length: 83cm  
Connection Interface: XH2.54-2P  
Colour: Black  
Other: Waterproof Probe

## Board Overview



Analog TDS Sensor / Meter For Arduino



Num	Label	Description
1	-	Power GND(0V)
2	+	Power VCC(3.3 ~ 5.5V)
3	A	Analog Signal Output(0 ~ 2.3V)
4	TDS	TDS Probe Connector
5	LED	Power Indicator

## Tutorial

This tutorial will show you how to measure the TDS value of the water. Please read this tutorial carefully, and pay attention to the steps and details.



The probe can not be used in water above 55 degrees centigrade.  
The probe can not be too close to the edge of the container, otherwise it will affect the reading.  
The head and the cable of the probe are waterproof, but the connector and the signal transmitter board are not waterproof. Please pay attention to use.

## Requirements

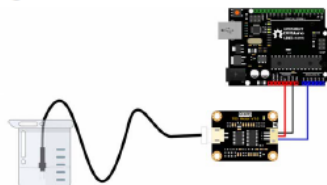
- **Hardware**  
DFRduino UNO R3 (or similar) x 1  
Analog TDS Sensor / Meter Module x 1  
TDS Probe x1  
Jumper Wires x3  
tested liquid x1

- **Software**

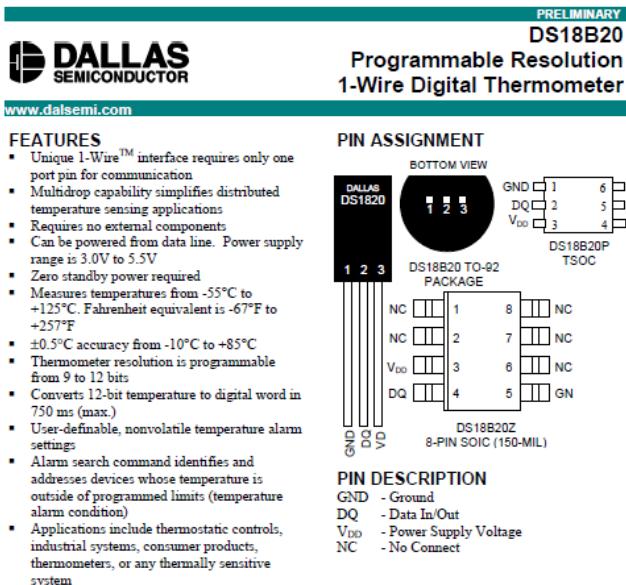
Arduino IDE (Version requirements: V1.0.x or V1.8.x), Click to Download Arduino IDE from Arduino®

<https://www.arduino.cc/en/Main/Software%7C>

## Connection Diagram



## A. Datasheet Sensor DS18B20



### DESCRIPTION

The DS18B20 Digital Thermometer provides 9 to 12-bit (configurable) temperature readings which indicate the temperature of the device.

Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.

DETAILED PIN DESCRIPTION Table 1

PIN TSOC	PIN 8PIN SOIC	PIN TO92	SYMBOL	DESCRIPTION
1	5	1	GND	Ground.
2	4	2	DQ	Data Input/Output pin. For 1-Wire operation: Open drain. (See "Parasite Power" section.)
3	3	3	V <sub>DD</sub>	Optional V <sub>DD</sub> pin. See "Parasite Power" section for details of connection. V <sub>DD</sub> must be grounded for operation in parasite power mode.

DS18B20Z (8-pin SOIC) and DS18P20P (TSOC): All pins not specified in this table are not to be connected.

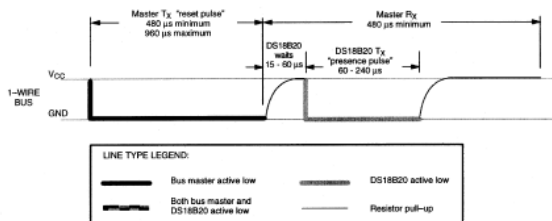
## OVERVIEW

The block diagram of Figure 1 shows the major components of the DS18B20. The DS18B20 has four main data components: 1) 64-bit lasered ROM, 2) temperature sensor, 3) nonvolatile temperature alarm triggers TH and TL, and 4) a configuration register. The device derives its power from the 1-Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS18B20 may also be powered from an external 3V - 5.5V supply.

Communication to the DS18B20 is via a 1-Wire port. With the 1-Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64-bit lasered ROM portion of each device and can single out a specific device if many are present on the 1-Wire line as well as indicate to the bus master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS18B20 to perform a temperature measurement. The result of this measurement will be placed in the DS18B20's scratch-pad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of 1 byte EEPROM each. If the alarm search command is not applied to the DS18B20, these registers may be used as general purpose user memory. The scratchpad also contains a configuration byte to set the desired resolution of the temperature to digital conversion. Writing TH, TL, and the configuration byte is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

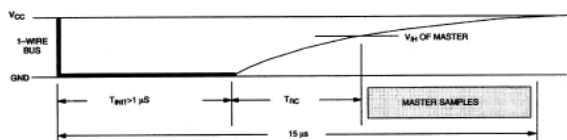
## INITIALIZATION PROCEDURE "RESET AND PRESENCE PULSES" Figure 11



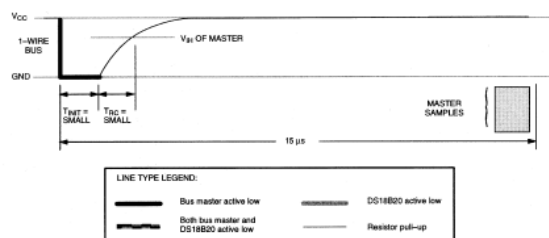
DS18B20 COMMAND SET Table 4

INSTRUCTION	DESCRIPTION	PROTOCOL	1-WIRE BUS AFTER ISSUING PROTOCOL	NOTES
<b>TEMPERATURE CONVERSION COMMANDS</b>				
Convert T	Initiates temperature conversion.	44h	<read temperature busy status>	1
<b>MEMORY COMMANDS</b>				
Read Scratchpad	Reads bytes from scratchpad and reads CRC byte.	BEh	<read data up to 9 bytes>	
Write Scratchpad	Writes bytes into scratchpad at addresses 2 through 4 (TH and TL temperature triggers and config).	4Eh	<write data into 3 bytes at addr. 2 through 4>	3
Copy Scratchpad	Copies scratchpad into nonvolatile memory (addresses 2 through 4 only).	48h	<read copy status>	2
Recall E <sup>2</sup>	Recalls values stored in nonvolatile memory into scratchpad (temperature triggers).	B8h	<read temperature busy status>	
Read Power Supply	Signals the mode of DS18B20 power supply to the master.	B4h	<read supply status>	

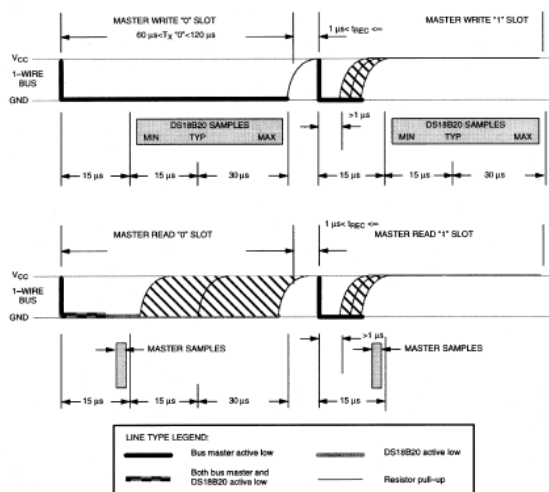
DETAILED MASTER READ 1 TIMING Figure 13



RECOMMENDED MASTER READ 1 TIMING Figure 14



READ/WRITE TIMING DIAGRAM Figure 12



## LAMPIRAN B

### A. Listing Program pada NodeMCU

```
#include "ESP8266WiFi.h"
#include <FirebaseArduino.h>
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR 0x3F // Jika 0x3F tidak mau coba
gunakan 0x27 atau 0x20
#define BACKLIGHT_PIN 3
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7
// Set these to run example.
#define FIREBASE_HOST "hydroponic-
8d669.firebaseio.com"
#define FIREBASE_AUTH
"VNO8zadrNDwngBICYHLD691HwnfhvjiOOXceESjR"
#define WIFI_SSID "gg"
#define WIFI_PASSWORD "gilanggg"

String data1, data2, data3, data4, incomingStr;
char kirim1[8], kirim2[8], kirim3[10], kirim4[10], kirim[40];
float ph, tds, lvl, temp, setphup, setphdown, settdsup,
settdsdown;
String readString;
char IncomingData[50];
LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_
pin,D7_pin);
void setup() {
  Serial.begin(115200);
  lcd.begin (20,4);
```

```

// Nyalakan lampu backlight
lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
lcd.setBacklight(HIGH);

// // connect to wifi.
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("connecting");
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}
Serial.println();
Serial.print("connected: ");
Serial.println(WiFi.localIP());

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

}

void loop() {
  lcd.clear();
  setphup = Firebase.getFloat("SetPHup");
  dtostrf(setphup, 2, 2, kirim1);
  setphdown = Firebase.getFloat("SetPHdown");
  dtostrf(setphdown, 2, 2, kirim2);
  settdsup = Firebase.getFloat("SetTDSup");
  dtostrf(settdsup, 4, 2, kirim3);
  settdsdown = Firebase.getFloat("setTDSdown");
  dtostrf(settdsdown, 4, 2, kirim4);
  //if(setphup != 0 && setphdown != 0 && settdsdown !=0)
  //{
  sprintf(kirim,"p%5s%5s%7s",kirim1,kirim2,kirim3);
  Serial.write(kirim);
  //}
  lcd.setCursor ( 0, 0 );           // tulis pada baris pertama
  lcd.print("pH      =");
  lcd.setCursor ( 0, 1 );
  lcd.print("TDS      =");
  lcd.setCursor ( 0, 2 );

```



```

lcd.print("Volume  =");
  lcd.setCursor ( 0, 3 );
  lcd.print("Temperatur =");
  while (Serial.available() > 0)
  {
    while(!Serial.available()); // wait till data to be filled in
    serial buffer

    incommingStr = Serial.readStringUntil("\n"); // read the
    complete string

    data1 = incommingStr.substring(0, 5);
    ph = data1.toFloat();
    data2 = incommingStr.substring(5, 12);
    tds = data2.toFloat();
    data3 = incommingStr.substring(12, 17);
    lvl = data3.toFloat();
    data4 = incommingStr.substring(17, 24);
    temp = data4.toFloat();

    break;
  }
  // Serial.println(incommingStr);
  // Serial.println(ph);
  // Serial.println(tds);
  // Serial.println(lvl);
  // Serial.println(temp);
  if(ph!=0 && tds !=0 && lvl !=0 && temp!=0)
  {
    lcd.setCursor ( 13, 0);
    lcd.print(ph);
    lcd.setCursor ( 13, 1);
    lcd.print(tds);
    lcd.setCursor ( 13, 2);
    lcd.print(lvl);
    lcd.setCursor ( 13, 3);
    lcd.print(temp);
    Firebase.setFloat("PH",ph);

```

```

    Firebase.setFloat("TDS",tds);
    Firebase.setFloat("Water_Level",lvl);
    Firebase.setFloat("Temperature",temp);
    if (Firebase.failed()) {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("setting /message failed:");
        lcd.setCursor(0,1);
        lcd.print(Firebase.error());
        return;
    }
}

}

```

#### **B. Listing Program pada STM32 F446RE NUCLEO**

```

#include "main.h"

/* Private includes -----
*/
/* USER CODE BEGIN Includes */
#include "dwt_stm32_delay.h"
#include "string.h"
#include "stdio.h"
#include "stdlib.h"
#include "i2c-lcd.h"
#include "stdbool.h"

int data_adc[3],temp[251], rataph, ratatds, ratalevel,
timetds,timephup,timephdown;
char kirim[50];
float nilai[4], sensor[4], miliphup, miliphdown, militds, stds, sphup,
sphdown;
int flag = 0;
int flagph = 0;
int x = 0;
int y = 0;
int counter = 0;
float vol = 9.48;

```

```

char data1[5],data2[5],data3[7];
uint8_t terima[18];
bool flagt = false;

GPIO_InitTypeDef GPIO_initStruct;
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    HAL_UART_Receive_IT(&huart3, terima, sizeof(terima));
}

void gpio_set_input (void)
{
    GPIO_initStruct.Pin = GPIO_PIN_1;
    GPIO_initStruct.Mode = GPIO_MODE_INPUT;
    GPIO_initStruct.Pull = GPIO_PULLUP;
    HAL_GPIO_Init(GPIOA, &GPIO_initStruct);
}

void gpio_set_output (void)
{
    GPIO_initStruct.Pin = GPIO_PIN_1;
    GPIO_initStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_initStruct.Pull = GPIO_NOPULL;
    GPIO_initStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOA, &GPIO_initStruct);
}
uint8_t check =2, temp_l, temp_h;
uint16_t temper;
float temperature;

uint8_t ds18b20_init (void)
{
    gpio_set_output ();
    HAL_GPIO_WritePin (GPIOA, GPIO_PIN_1, 0);
    DWT_Delay_us (480);
}

```

```

gpio_set_input ();
DWT_Delay_us (80);

if (!(HAL_GPIO_ReadPin (GPIOA, GPIO_PIN_1)))
{
DWT_Delay_us (400);
return 0;
}

else
{
DWT_Delay_us (400);
return 1;
}
}

void write (uint8_t data)
{
gpio_set_output (); // set as output

for (int i=0; i<8; i++)
{

if ((data & (1<<i))!=0) // if the bit is high
{
// write 1

gpio_set_output (); // set as output
HAL_GPIO_WritePin (GPIOA, GPIO_PIN_1, 0); // pull the pin
LOW
DWT_Delay_us (1); // wait for us

gpio_set_input (); // set as input
DWT_Delay_us (60); // wait for 60 us
}

else // if the bit is low
{
// write 0

```

```

gpio_set_output ();
HAL_GPIO_WritePin (GPIOA, GPIO_PIN_1, 0); // pull the pin
LOW
DWT_Delay_us (60); // wait for 60 us

gpio_set_input ();
}
}
}

uint8_t read (void)
{
uint8_t value=0;
gpio_set_input ();

for (int i=0;i<8;i++)
{
gpio_set_output (); // set as output

HAL_GPIO_WritePin (GPIOA, GPIO_PIN_1, 0); // pull the data
pin LOW
DWT_Delay_us (2); // wait for 2 us

gpio_set_input (); // set as input
if (HAL_GPIO_ReadPin (GPIOA, GPIO_PIN_1)) // if the pin is
HIGH
{
value |= 1<<i; // read = 1
}
DWT_Delay_us (60); // wait for 60 us
}
return value;
}

int main(void)
{
HAL_Init();
MX_GPIO_Init();

```

```

MX_DMA_Init();
MX_ADC1_Init();
MX_TIM1_Init();
MX_USART3_UART_Init();
MX_I2C1_Init();

HAL_ADC_Start_DMA(&hadc1,(uint32_t*)data_adc,3);
HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);
HAL_TIM_Base_Start_IT(&htim1);
}
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */

    HAL_UART_Receive_IT(&huart3, terima, sizeof(terima));
    if(terima[0]=='p')
    {

        memcpy(data1,terima+1,5);

        sphup = atof(data1);

        memcpy(data2,terima+6,5);

        sphdown = atof(data2);

        memcpy(data3,terima+11,7);

        stds = atof(data3);
    }

    HAL_GPIO_TogglePin(LD2_GPIO_Port,LD2_Pin);

    //float rumus= 0.0016*data_adc[0];
    rataph = 0;
    ratatds = 0;

```

```

    ratalevel = 0;
// Sensor pH
    for(int i=0 ; i<250 ; i++)
    {

temp[i] = data_adc[0];

    rataph = temp[i]+rataph;

    HAL_Delay(2);
    }
    nilai[0] = rataph/250;
    sensor[0] = ((-2e-08)*nilai[0]*nilai[0]) + (0.0033*nilai[0])+1.9873;
    x++;
    y++;
// Sensor TDS
    for(int i=0 ; i<250 ; i++)
    {

temp[i] = data_adc[1];

    ratatds = temp[i]+ratatds;

    HAL_Delay(2);
    }
    nilai[1] = ratatds/250;
    sensor[1] = ((8e-05)*nilai[1]*nilai[1]) + (0.2731*nilai[1])+27.936;
    x++;
    y++;
// Sensor Water Level
    for(int i=0 ; i<250 ; i++)
    {

temp[i] = data_adc[2];
    ratalevel = temp[i]+ratalevel;
    HAL_Delay(2);
    }
    nilai[2] = ratalevel/250;
    sensor[2] = nilai[2]/99.8;

```

```

x++;
y++;

// Sensor TemperaturDS18B20

check = ds18b20_init ();
write (0xCC);
write (0x44);
HAL_Delay (800);
ds18b20_init ();
write (0xCC);
write (0xBE);
temp_l = read();
temp_h = read();
temper = (temp_h<<8)|temp_l;
sensor[3] = (float)temper/16;

sprintf(kirim,"%5.2f%7.2f%5.2f%7.2f\n",sensor[0],sensor[1],sensor[
2],sensor[3]);

HAL_UART_Transmit(&huart3,(uint8_t*)kirim,sizeof(kirim),50);

HAL_Delay(100);
//TIM1->CCR1=128;
// Kontrol
militds = (0.01681*vol)+(-0.1595*sensor[1])+(0.1655*stds);
if (militds<0){ militds=militds*(-1);}
miliphdown = (-1.18*vol)+(2.58*sensor[0])+(-0.7*sphup);
miliphup = (0.65*vol)+(0.87*sensor[0])+(1.92*sphdown);
timephup = ((1.303*miliphup+1.0364)*1000);
timephdown = ((1.303*miliphdown+1.0364)*1000);

//inject phUp
(sensor[0]<=sphdown && flagph == 1 && timephup !=0)
{
    if (timephup<0){timephup=timephup*(-1);}

HAL_GPIO_WritePin(GPIOB, pHUp_Pin, GPIO_PIN_SET);

```



```

HAL_Delay(timephup);
y=0;
flagph=0;
}
else
{
HAL_GPIO_WritePin(GPIOB, pHUp_Pin, GPIO_PIN_RESET);
}

// inject phdown

if (sensor[0]>=sphup && flagph == 1)

{

HAL_GPIO_WritePin(GPIOB, pHDown_Pin, GPIO_PIN_SET);
HAL_Delay(timephdown);
y=0;
flagph=0;
}

else
{
HAL_GPIO_WritePin(GPIOB, pHDown_Pin, GPIO_PIN_RESET);
}

// inject nutrisi a dan nutrisi b

if (sensor[1]<=stds && flag == 1)
{
timetds = ((1.303*militds+1.0364)*1000)/2;
HAL_GPIO_WritePin(GPIOC, NUTRISIA_Pin, GPIO_PIN_SET

HAL_Delay(timetds);
HAL_GPIO_WritePin(GPIOC, NUTRISIB_Pin, GPIO_PIN_SET);
HAL_Delay(timetds);
timetds = 0;
x=0;

```

```

    flag=0;

}

else
{
    HAL_GPIO_WritePin(GPIOC, NUTRISIA_Pin,
    GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOC, NUTRISIB_Pin,
    GPIO_PIN_RESET);
}

if(x>=120)
{
    x = 125;

    flag = 1;
}
    if(y>=200)
    {
        y = 208;
        flagph = 1;
    }
}

```

### C. Listing Program Website HTML

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>SI Hidroponik</title>
    <link rel="stylesheet"
href="vendor/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="vendor/font-awesome/css/font-
awesome.min.css">
    <link rel="stylesheet" href="css/font.css">

```

```

<link rel="stylesheet" href="css/style.default.css" id="theme-
stylesheet">
<link rel="shortcut icon" href="img/favicon.ico">
<!-- JavaScript files-->
<script src="vendor/jquery/jquery.min.js"></script>
<!-- <script src="vendor/bootstrap/js/bootstrap.min.js"></script> -
->
<script src="vendor/chart.js/Chart.min.js"></script>
<script src="js/charts-home.js"></script>
<!-- <script src="js/front.js"></script> -->
<script
src="https://www.gstatic.com/firebasejs/5.4.1/firebase.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js
"></script>
<link href="https://gitcdn.github.io/bootstrap-
toggle/2.2.2/css/bootstrap-toggle.min.css" rel="stylesheet">
<script src="https://gitcdn.github.io/bootstrap-
toggle/2.2.2/js/bootstrap-toggle.min.js"></script>
<script>

```

```

var config = {
  apiKey:
"AIzaSyA2WHA1RVjeBaiD3PV3DoQBge3s8UAhBLA",
  authDomain: "hydroponic-8d669.firebaseio.com",
  databaseURL: "https://hydroponic-8d669.firebaseio.com",
  projectId: "hydroponic-8d669",
  storageBucket: "hydroponic-8d669.appspot.com",
  messagingSenderId: "357523484402"
};
// document.getElementById("coba").innerText = "hallo";
firebase.initializeApp(config);
//push database
var d = new Date();
var date =d.getDate();
var month =d.getMonth();
var year =d.getFullYear();
var h = d.getHours();
var m = d.getMinutes();

```

```

var s = d.getSeconds();
var time = date+"-"+month+"-"+year+" "+h+": "+m+": "+s;
var playersRef = firebase.database().ref().child("players");
window.onload = function(){ playersRef.push ({
    date: time,
    temp: Water_Level,
    ph: 6,
    tds: 60,
    lvl: 66
});
}

```

```

//login PAGE
firebase.auth().onAuthStateChanged(function(user){
    if (user){
        //window.location.href = "index.html"
    }
    else{
        window.location.href = "login.html"
    }
});
//logout PAGE
function logout(){
    firebase.auth().signOut().then(function() {
        // Sign-out successful.
    }).catch(function(error) {
        // An error happened.
    });
}
//write data to firebase
while(0){
    var phc = document.getElementById('switch').checked;
    if(phc==true){
        firebase.database().ref().child('controlph').set(0);
    }
    else{
        firebase.database().ref().child('controlph').set(1);
    }
}

```

```

    }
    document.getElementById('coba').innerText = phc;
    break;
}

function setPHup(){
    var fb = firebase.database();
    var PHref = fb.ref('SetPHup');
    var Set_pH = document.getElementById('InputpHUp').value;
    var phfloat = parseFloat(Set_pH);
    var result = PHref.set(phfloat);
    var inputPH = document.getElementById('InputpHUp');
    inputPH.value = "";
}

function setPHdown(){
    var fb = firebase.database();
    var PHref = fb.ref('SetPHdown');
    var Set_pH =
document.getElementById('InputpHDwon').value;
    var phfloat = parseFloat(Set_pH);
    var result = PHref.set(phfloat);
    var inputPH = document.getElementById('InputpHDwon');
    inputPH.value = "";
}

function setTDSup(){
    var fb = firebase.database();
    var TDSref = fb.ref('SetTDSup');
    var Set_TDS =
document.getElementById('InputTDSUp').value;
    var TDSfloat = parseFloat(Set_TDS);
    var result = TDSref.set(TDSfloat);
    var inputTDS = document.getElementById('InputTDSUp');
    inputTDS.value = "";
}

function setTDSdown(){
    var fb = firebase.database();

```

```

    var TDSref = fb.ref('SetTDSdown');
    var Set_TDS =
document.getElementById('InputTDSDown').value;
    var TDSfloat = parseFloat(Set_TDS);
    var result = TDSref.set(TDSfloat);
    var inputTDS = document.getElementById('InputTDSDown');
    inputTDS.value = "";
}

//read data from firebase
var f1 = firebase.database().ref().child("PH");
var f1a = firebase.database().ref().child("SetUpPH");
var f2 = firebase.database().ref().child("TDS");
var f2a = firebase.database().ref().child("SetUpTDS");
var f3 = firebase.database().ref().child("Temperature");
var f4 = firebase.database().ref().child("WaterLevel");

var x;

var phval,tdsval;
f1.on('value',function(datanapshot){
    SensorPH.innerText = datanapshot.val();
    myBarPH.style.width = (datanapshot.val()/14)*100+"%";
})
f1a.on('value',function(datanapshot){
    phval.innerText = datanapshot.val();
})
f2.on('value',function(datanapshot){
    SensorTDS.innerText = datanapshot.val();
    myBarTDS.style.width = (datanapshot.val()/2000)*100+"%";
})
f2a.on('value',function(datanapshot){
    TDSval.innerText = datanapshot.val();
})
f3.on('value',function(datanapshot){
    Temp.innerText = datanapshot.val()+"°C";
    myBarTemp.style.width = (datanapshot.val()/100)*100+"%";
})

```

```
f4.on('value',function(datasnapshot){
    Water.innerText = datasnapshot.val()+"%";
    myBarWater.style.width = (datasnapshot.val()/100)*100+"%";
})
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<header class="header">
```

```
<nav class="navbar navbar-expand-lg">
```

```
<div id="coba"></div>
```

```
<div class="container-fluid d-flex align-items-center justify-
content-between">
```

```
<div class="navbar-header">
```

```
<!-- Navbar Header-->
```

```
<a href="index.html" class="navbar-brand">
```

```
<div class="brand-text brand-big visible text-
```

```
uppercase"><strong class="text-
```

```
primary">HIDRO</strong><strong>PONIK</strong></div>
```

```
<div class="brand-text brand-sm"><strong class="text-
```

```
primary">H</strong><strong>P</strong></div></a>
```

```
<!-- Sidebar Toggle Btn-->
```

```
<button class="sidebar-toggle"><i class="fa fa-long-arrow-
left"></i></button>
```

```
</div>
```

```
<div class="right-menu list-inline no-margin-bottom">
```

```
<div class="list-inline-item"><a href="#" class="search-open
nav-link"><i class="icon-magnifying-glass-
browser"></i></a></div>
```

```
<!-- Log out -->
```

```
<div class="list-inline-item logout">
```

```
<a
id="logout" onclick="logout()" href="login.html" class="nav-link">
<span class="d-none d-sm-inline">Logout </span><i class="icon-
logout"></i></a></div>
```

```
</div>
```

```
</div>
```

```

</nav>
</header>
<div class="d-flex align-items-stretch">
  <!-- Sidebar Navigation-->
  <nav id="sidebar">
    <!-- Sidebar Header-->
    <div class="sidebar-header d-flex align-items-center">
      <div class="avatar"></div>
      <div class="title">
        <h1 class="h5">Tunas Urban Farm</h1>
        <p>Perusahaan</p>
      </div>
    </div>
    <!-- Sidebar Navigation Menus--><span
class="heading">Main</span>
    <ul class="list-unstyled">
      <li class="active"><a href="index.html"> <i class="icon-
home"></i>Home </a></li>
      <li><a href="tables.html"> <i class="icon-grid"></i>Tables
</a></li>
      <li><a href=""> <i class="fa fa-bar-chart"></i>Charts
</a></li>
      <li><a href=""> <i class="icon-padnote"></i>Forms </a></li>
      <li><a href="login.html"> <i class="icon-logout"></i>Login
page </a></li>
    </ul>
  </nav>
  <!-- Sidebar Navigation end-->
  <div class="page-content">
    <div class="page-header">
      <div class="container-fluid">
        <h2 class="h5 no-margin-bottom">Dashboard</h2>
      </div>
    </div>
    <div class="page-control">
      <div class="container-fluid">
        <div class="row">
          <h2 class="h3 col-sm-2">Control pH</h2>

```



```

        <div class="col">
            <input type="checkbox" id="switch" /><label
for="switch">Toggle</label>
        </div>
    </div>
</div>
</div>
</div>
<section class="no-padding-top no-padding-bottom">
    <div class="container-fluid">
        <div class="row">
            <div class="col-md-3 col-sm-6">
                <div class="statistic-block block">
                    <div class="progress-details d-flex align-items-end
justify-content-between">
                        <div class="title">
                            <div class="icon"><i class="icon-
flask"></i></div><strong>pH</strong>
                        </div>
                        <div id="SensorPH" class="number dashtext-1"></div>
                    </div>
                    <div class="progress progress-template">
                        <div id="myBarPH" role="progressbar" aria-
valuenow="30" aria-valuemin="0" aria-valuemax="100"
class="progress-bar progress-bar-template dashbg-1"></div>
                    </div>
                </div>
            </div>
            <div class="col-md-3 col-sm-6">
                <div class="statistic-block block">
                    <div class="progress-details d-flex align-items-end
justify-content-between">
                        <div class="title">
                            <div class="icon"><i class="icon-
contract"></i></div><strong>TDS</strong>
                        </div>
                        <div id="SensorTDS" class="number dashtext-
3"></div>
                    </div>
                    <div class="progress progress-template">

```

```

        <div id="myBarTDS" role="progressbar" aria-
valuenow="70" aria-valuemin="0" aria-valuemax="100"
class="progress-bar progress-bar-template dashbg-3"></div>
    </div>
</div>
</div>
<div class="col-md-3 col-sm-6">
    <div class="statistic-block block">
        <div class="progress-details d-flex align-items-end
justify-content-between">
            <div class="title">
                <div class="icon"><i class="icon-paper-and-
pencil"></i></div><strong>Temperature</strong>
            </div>
            <div id="Temp" class="number dashtext-2"></div>
        </div>
        <div class="progress progress-template">
            <div id="myBarTemp" role="progressbar" aria-
valuenow="55" aria-valuemin="0" aria-valuemax="100"
class="progress-bar progress-bar-template dashbg-2"></div>
        </div>
    </div>
</div>
<div class="col-md-3 col-sm-6">
    <div class="statistic-block block">
        <div class="progress-details d-flex align-items-end
justify-content-between">
            <div class="title">
                <div class="icon"><i class="icon-writing-
whiteboard"></i></div><strong>Water Level</strong>
            </div>
            <div id="Water" class="number dashtext-4"></div>
        </div>
        <div class="progress progress-template">
            <div id="myBarWater" role="progressbar" aria-
valuenow="35" aria-valuemin="0" aria-valuemax="100"
class="progress-bar progress-bar-template dashbg-4"></div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
</div>
</section>
<section class="no-padding-bottom">
    <div class="container-fluid">
        <div class="row">
            <div class="col-lg-4">
                <div class="row">
                    <div class="block">
                        <div id="setph" class="title"><strong>Setting Setpoint
pH</strong></div>
                        <div class="block-body">
                            <form class="form-inline">
                                <div class="form-group">
                                    <input id="InputpHUp" type="number"
placeholder="Set-point Above" class="mr-sm-3 form-control">
                                </div>
                                <div class="form-group">
                                    <input type="submit" onclick="setPHup()"
value="Submit" class="btn btn-secondary">
                                </div>
                                <div class="form-group">
                                    <input id="InputpHDwon" type="text"
placeholder="Set-Point Bottom" class="mr-sm-3 form-control form-
control">
                                </div>
                                <div class="form-group">
                                    <input type="submit" onclick="setPHdown()"
value="Submit" class="btn btn-secondary">
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
            <div class="block">
                <div class="title"><strong>Setting Setpoint
TDS</strong></div>

```

```

        <div class="block-body">
            <form class="form-inline">
                <div class="form-group">
                    <input id="InputTDSUp" type="text"
placeholder="Set-point Above" class="mr-sm-3 form-control">
                </div>
                <div class="form-group">
                    <input type="submit" onclick="setTDSUp()"
value="Submit" class="btn btn-primary">
                </div>
                <div class="form-group">
                    <input id="InputTDSDown" type="text"
placeholder="Set-Point Bottom" class="mr-sm-3 form-control form-
control">
                </div>
                <div class="form-group">
                    <input type="submit" onclick="setTDSdown()"
value="Submit" class="btn btn-primary">
                </div>
            </form>
        </div>
    </div>
</div>
</div>
</div>
<div class="col-lg-8">
    <div class="line-cahrt block">
        <canvas id="lineCahrt"></canvas>
    </div>
</div>
</div>
</div>
</div>
</section>
<footer class="footer">
    <div class="footer__block block no-margin-bottom">
        <div class="container-fluid text-center">
            <p class="no-margin-bottom"></p>
        </div>
    </div>
</footer>





```

```
</div>  
</div>  
</body>  
</html>
```

-----Halaman ini sengaja dikosongkan-----

## LAMPIRAN C

### A. Gambar Pengukuran Sensor dan Alat Ukur

Pengukuran Sensor	Pengukuran Alat Ukur
	
	





## DAFTAR RIWAYAT HIDUP



Nama : Gilang Sasmito  
TTL : Madiun, 14  
Februari 1998  
Jenis Kelamin : Laki-Laki  
Agama : Islam  
Alamat Rumah : Ds. Bacem Kec.  
Kebonsari Kab.  
Madiun  
Telpn/HP : 085607436718  
E-mail :  
sasmitogilang@gmail.com

### RIWAYAT PENDIDIKAN

1. 2002 – 2004 : TK Tunas Bangsa Kendari
2. 2004 – 2010 : SD Bacem Madiun
3. 2010 – 2013 : SMPN 1 Geger Madiun
4. 2013 – 2016 : SMA
5. 2015 – 2018 : Departemen Teknik Elektro  
Otomasi Program Studi Elektro  
Industri - Fakultas Vokasi (FV)  
Institut Teknologi Sepuluh  
Nopember (ITS)

### PENGALAMAN KERJA

1. Magang di CV. Panca Gourmet Nusantara September – Januari 2019

### PENGALAMAN ORGANISASI

1. Staff *Line Tracer Analog* - IARC
2. Staff Ahli *Line Tracer Analog* - IARC

-----Halaman ini sengaja dikosongkan-----